INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Analysis of random trees

Carlos Alberto Zamora Cura

School of Computer Science

McGill University, Montreal

January, 2000

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of PhD in Science

Abstract

In the first part of the thesis, we analyze the expected time complexity of range searching with k-d trees in all dimensions when the data points are uniformly distributed in the unit hypercube. The partial match results of Flajolet and Puech are reproved using elementary probabilistic methods. In addition, we analyze the expected complexity of orthogonal and convex range search, as well as nearest neighbor search. We introduce a new data structure, the squarish k-d tree, in which the longest edge is always cut first. This modification makes the expected time behavior of lower-dimensional partial match queries behave as for perfectly balanced complete k-d trees on n nodes. This is in contrast to a result of Flajolet and Puech, who proved that for (standard) random k-d trees with cuts that rotate among the coordinate axes, the expected time behavior is much worse than for balanced complete k-d trees. We show that the expected complexity for range search and nearest neighbor search for squarish k-d trees is either optimal or near optimal.

In the second part, we analyze branch-and-bound search for random b-ary trees. In particular, let T_n be a b-ary tree of height n, that has independent, nonnegative, identically distributed random variables associated with each of its edges. The value of a node is the sum of all the edge values on its path to the root. Consider the problem of finding the minimum leaf value of T_n. Assume that the edge random variable X is nondegenerate, has $\mathbf{E}\{X^\theta\} < \infty$ for some $\theta > 2$, and satisfies $bP\{X = c\} < 1$ where c is the left-most point of the support of X. We analyze the performance of the standard branch-and-bound algorithm for this problem and prove that the number of nodes visited is in probability $(\beta + o(1))^n$, where $\beta \in (1, b)$ is a constant

depending only on the distribution of the edge random variables. Explicit expressions for β are derived. We also show that any search algorithm must visit $(\beta + o(1))^n$ nodes with probability tending to one, so branch-and-bound is asymptotically optimal where first-order asymptotics are concerned.

Résumé

Dans la première partie de la thèse, on analyse l'espérance du temps de complexité de la recherche étendue avec des arbres k-d pour toutes les dimensions où les données sont uniformément distribuées au sein de l'hypercube unité. Les résultats de concordance partielle de Flajolet et Puech sont démontrés á nouveau à l'aide de méthodes probabilistes élémentaires. De plus, on analyse l'espérance de la complexité de la recherche à champ orthogonal et convexe ainsi que de la recherche du voisin le plus proche. On introduit une nouvelle structure de données, l'arbre k-d squarish, dans lequel l'arête la plus longue est toujours la première coupée. Cette modification implique que l'espérance du temps de recherche à concordance partielle de basse dimension se comporte comme pour les arbres k-d complets à n sommets parfaitement équilibrés. Ceci contraste avec un résultat de Flajolet et Puech, qui ont prouvé que pour les arbres k-d (standards) avec des coupures qui alternent entre les axes de coordonnées, le comportement de l'espérance de temps est bien pire que pour les arbres k-d equilibrés. On démontre que l'espérance de la complexité pour la recherche étendue et celle du voisin le plus proche pour les arbres k-d squarish est soit optimale ou bien presqu'optimale.

Dans la seconde partie, on analyse la recherche branch-and-bound pour les arbres b-ary aléatoires. En particulier, soit T_n un arbre b-ary de hauteur n possédant des variables aléatoires associées à chacune de ses arêtes qui sont non-négatives, indépendantes, et identiquement distribuées. La valeur d'un sommet est la somme des valeurs de toutes les arêtes sur son chemin vers la racine. Condérons le problème qui consiste à trouver la valeur-feuille minimale de T_n. Admettons que la variable aléatoire d'arête X est

non-dégénérée et satisfait $\mathbf{E}\{X^\theta\} < \infty$ pour un $\theta > 2$, ainsi que $bP\{X = c\} < 1$, où c est le point du support de X le plus à gauche. On analyse la performance de l'algorithme branch-and-bound standard pour ce problème et l'on prouve que le nombre de sommets visités est en probabilité $(\beta+o(1))^n$, où $\beta \in (1, b)$ est une constante ne dèpendant que de la distribution des variables aléatoires d'arête. Des formules explicites pour β sont dérivées. On démontre aussi que tout algorithme de recherche doit visiter $(\beta + o(1))^n$ sommets avec une probabilité tendant vers un. Donc en autant que le premier ordre est concerné, branch-and-bound est asymptotiquement optimal.

Acknowledgements

First of all, I want to express my gratitude to Luc Devroye, my thesis supervisor. Right from the beginning, he has shared his ideas with me in an unselfish way. His enthusiasm, energy and perfect teaching were the main factors in getting me interested in probabilistic analysis of algorithms. Without his help and advice this thesis would have not been possible. My time at McGill has been much more enjoyable since I started my PhD under his supervision.

I received financial assistance from a scholarship DGAPA-UNAM for graduate studies. Many thanks again to Luc Devroye for his financial support.

My thanks also go to my parents for always believing in me. To my friends Carlos and Manuel for their company, both intellectual and emotional. To Refugio for pushing me to have some very needed exercise during my graduate studies.

Finally my deepest thanks to my wife Carmen for her patience, understanding and love. Without her my life would be a total mess.

To my tender wife
Carmen

Contents

Introduction

To analyze an algorithm, in a broad sense, means to characterize the amount of resources that an execution of the algorithm will require when applied to input data of a given length. There are several ways of making this definition more precise. We may want to know the worst case behavior of the algorithm with respect to the general resources the algorithm will need to perform its tasks: we want to have an absolute warranty that the algorithm will not use more than a certain amount of storage space or a given time complexity on any input of at most a given length.

The worst case measure is perhaps the most common and well-known approach when analyzing an algorithm. However, this approach may hide the typical behavior of the algorithm since the worst case input data may be rare among all possible inputs.

Another approach is to analyze the algorithm from a probabilistic perspective. We can do this in several ways. For example, in one approach, we assume that the input data is distributed according to some probability distribution. The amount of resources that the algorithm uses is quantified in a probabilistic sense. We may want to know the expected space complexity or the expected time complexity for instances of a given size. In general, we choose some probability distribution on the inputs of a given size and analyze the performance of the algorithm when applied to a random input drawn from this distributions.

Range Search and Nearest Neighbor Search

Data structures for multi-attribute data should support the usual dictionary operations as well as some associative queries. Examples of associative

queries are partial match, range search and nearest neighbor queries. A partial match query asks for all the elements in the file that match a given vector with possibly a number of wild-cards. The ancestry of most methods to solve partial match queries is to be found in works by Rivest (1976), where hashing and digital techniques are explored, and by Bentley (1975) and Bentley and Finkel (1974) who proposed k-d trees and quad trees, which are comparison based structures to solve these problems (see also Knuth, 1997).

Range search is a fundamental problem in many fields such as computational geometry, data base theory and pattern recognition. For a range search query a set of points is given and a search set is specified. The objective is to return the points that lie within the search region. Search regions may take several forms such as half spaces, simplex regions, convex regions or orthogonal ranges. Orthogonal range search is of interest to us. When orthogonal ranges (i.e., hyper-rectangles in the k dimensional Euclidean space) are considered, range trees (see, Bentley (1977), Preparata and Shamos (1985)) are among the best data structures, in a worst case scenario. To solve the problem, they have $O((\log n)^{k-1} + N)$ query time and use $O(n(\log n)^{k-1})$ units of space, for n data points, where N is the number of points in the query region. However, the implementation of this data structure is quite cumbersome, so simpler methods are of interest. k-d trees are binary trees that store multidimensional data. A k-d tree is built up so that at each level of the tree a specific component of the data is used for splitting. The components of the data are used cyclically on the path down the tree. For any node u having index $j \in \{1, \ldots, k\}$, all nodes in its left subtree are such that their j^{th} key is less than the j^{th} key in u, and all nodes in its right subtree are such that their j^{th} key is greater or equal than the j^{th} key in u. The root is assigned index 1, and from there on the index of each node is determined by the depth of the node in the tree in a rotational fashion. Insertion and search are implemented as for the standard binary search tree algorithms. The deletion procedure is a bit more complex than the deletion procedure in binary search trees as we must keep the cyclic order of the partition coordinates.

It nevertheless can be implemented in a very similar manner. The k-d tree offers several advantages–it takes $O(kn)$ space for n data points, it is easily updated and maintained, it is simple to implement and comprehend, and it is useful for other operations besides orthogonal range search. We can also implement range search by storing the data in a k-d tree by simply visiting recursively all subtrees of the root that have a nonempty intersection with the query rectangle.

We can also implement nearest neighbor search using k-d trees. Given a point x and a set of points in the plane, nearest neighbor search asks for the nearest neighbor of x among all the points in the given set. This problem has been studied extensively in areas such as computational geometry and pattern recognition, where nearest neighbor queries are of central importance.

Branch-and-Bound Search

Optimization of rooted trees is a routine problem in computer science and operations research. The study of efficient algorithms for finding the best leaf in a rooted tree is the subject of many projects and papers in the artificial intelligence community. If leaves have values associated with them and a minimum must be found, one may perform exhaustive search, branch-and-bound search (which is a depth-first search with on-line pruning of useless subtrees), backtracking, backtracking with bounded lookahead, and variations of these methods (Reingold, Nievergelt and Deo (1977), Kumar (1992)). To compare various methods, toy models have been proposed, of which the model of Karp and Pearl (1983) is perhaps the most interesting. Karp and Pearl consider a complete binary tree with n levels of edges and associate with each edge an independent Bernoulli(p) random variable. Each node v has a value V_v equal to the sum of the edge values on the path to the root. Each value V_v is available upon request, but each request costs one time unit. The objective is to find the leaf of minimal value. Karp and Pearl noted that if $2p > 1$, any algorithm must necessarily take exponential expected time in n, while

for $2p = 1$ and $2p < 1$, ordinary uniform cost breadth-first search takes on the average $\Theta(n^2)$ and $\Theta(n)$ time. In this algorithm, one first visits all nodes of value 0, then all nodes of value 1, and so forth. McDiarmid (1990) and McDiarmid and Provan (1990), generalized the work of Karp and Pearl to b-ary trees and more general distributions.

Outline of the thesis

In the first part of this thesis we analyze the expected time complexity of partial match search when using as underlying data structure k-dimensional trees. As we will see in the forthcoming chapters, the expected time complexity of range search can be computed if the expected time complexity of partial match queries are known. Flajolet and Puech (1986) computed the expected time complexity of partial match queries in the k-dimensional unit cube using generating function techniques. They found that the expected time complexity of partial match when s out of the k attributes are specified is asymptotic to $n^{1-s/k+\theta(s/k)}$, where $0 \leq \theta(u) \leq 0.07$ for $u \in [0,1]$. Thus, the expected time complexity of partial match is not optimal when using k-d trees, as off-line, if a median k-d tree is constructed from the data, any partial match query takes worst case time $O(n^{1-1/k}+N)$ where N is the number of points in the query region. We reprove the Flajolet and Puech (1986) results using probabilistic methods. This is work reported in Chanzy, Devroye and Zamora-Cura (1999).

Having this result as starting point, we analyze the expected time complexity of range search when using k-d trees. We also introduce the "squarish k-d tree", which is a k-dimensional tree that has optimal expected time complexity with respect to partial match queries, explicitly $\Theta(n^{1-s/k})$, when s out of k attributes have been specified. We also study the complexity of range search when using squarish k-d trees. These results can be found in Devroye, Jabbour and Zamora-Cura (1999).

Using the results about range search we analyze two algorithms to solve the nearest neighbor problem. In the first algorithm, range search queries centered at the query point are performed on boxes of growing side length. We return the nearest neighbor of the point in the second nonempty box. We analyze both k-d trees and squarish k-d trees with respect to this algorithm. The second algorithm works by returning the nearest neighbor among all the points lying in the box of sides of length twice the perimeter of the box on which the query point lies.

In the second part of this thesis, we generalize and complement the results of Karp and Pearl (1983), McDiarmid (1990) and McDiarmid and Provan (1990). We show that in probability, the number of nodes visited by the standard branch-and-bound algorithm to find the minimum leaf value in a random tree (as introduced in the previous section) is $(\beta + o(1))^n$, where $\beta \in (1, b)$ is a constant depending only on the distribution of the edge random variables, and b is the fan-out of the tree and n is the number of levels. We derive explicit expressions for β. We also show that any search algorithm must visit $(\beta + o(1))^n$ nodes with probability tending to one, so branch-and-bound is asymptotically optimal is a first-order asymptotic sense. These chapters are based on Devroye and Zamora-Cura (1999).

Chapter 1
Preliminaries

In this chapter we present the basic mathematical tools that we will be using through out the thesis.

1.1 Probability

DEFINITION. *A σ-algebra $(\Omega, \mathcal{F})$ consists of a sample space Ω and a collection of subsets $\mathcal{F}$ of Ω satisfying that,*

- $\emptyset \in \mathcal{F}$,
- *if $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$, and*
- $\forall \{A_i\}_{i \in \mathbf{N}}$ *of sets in* $\mathcal{F}$, $\cup_{i \in \mathbf{N}} A_i \in \mathcal{F}$.

The first and second conditions imply that $\Omega \in \mathcal{F}$. Moreover, the second and third conditions imply that a σ-algebra is closed under countably infinite intersections. If $\Omega = \mathbb{R}^k$ and $\mathcal{B}$ is the smallest σ-algebra containing all the rectangles, then $\mathcal{B}$ is called the Borel σ-algebra. The elements of $\mathcal{B}$ are called Borel sets.

DEFINITION. *A probability space is a triple $(\Omega, \mathcal{F}, \mathbf{P})$ where $(\Omega, \mathcal{F})$ is a σ-algebra, and $\mathbf{P}$ is a function from $\mathcal{F}$ to $[0, 1]$ such that,*

- $\mathbf{P}\{\emptyset\} = 0$,
- $\mathbf{P}\{\cdot\}$ *is σ-additive, this is $A_1, A_2, \ldots \in \mathcal{F}$ and $A_i \cap A_j = \emptyset$, $i \neq j$, imply that* $\mathbf{P}\{\cup_{i=1}^{\infty} A_i\} = \sum_{i=1}^{\infty} \mathbf{P}\{A_i\}$,

The sets in $\mathcal{F}$ are called *events*. When the set Ω is finite the σ-algebra considered is its power set. In this thesis the measurability questions are irrelevant as the standard σ-algebras are rich enough so as to avoid any type of measurability problem.

THEOREM 1.1. (BOOLE'S INEQUALITY). *Let* $(\Omega, \mathcal{F}, \mathbf{P})$ *be a probability space. If* $A_1, \ldots, A_n \in \mathcal{F}$ *then,*

$$\mathbf{P}\left\{\cup_{i=1}^{n} A_i\right\} \leq \sum_{i=1}^{n} \mathbf{P}\left\{A_i\right\}.$$

Given a probability space $(\Omega, \mathcal{F}, \mathbf{P})$, and $A, B \in \mathcal{F}$, with $\mathbf{P}\{B\} > 0$, we define the *conditional probability of A given B*, $\mathbf{P}\{A|B\}$, as

$$\frac{\mathbf{P}\{A \cap B\}}{\mathbf{P}\{B\}}.$$

It can easily be verified that conditional probabilities are indeed probability functions. A *random variable* X over $(\Omega, \mathcal{F}, \mathbf{P})$ is a real-valued function over the sample space, $X : \Omega \to \mathbb{R}$, such that for all $x \in \mathbb{R}$,

$$\{\omega \in \Omega : X(\omega) \leq x\} \in \mathcal{F}.$$

This definition allows us to have a more compact notation for complex events since, for example, $\mathbf{P}\{\omega \in \Omega : X(\omega) \leq x\}$ becomes $\mathbf{P}\{X \leq x\}$.

Given a random variable X over $(\Omega, \mathcal{F}, \mathbf{P})$ we define its *expected value* as follows,

$$\mathbf{E}\{X\} = \int_{\Omega} X d\mathbf{P}.$$

When the random variable is discrete the expected value of X becomes $\sum_{\omega \in \Omega} X(\omega)\, \mathbf{P}\{X = \omega\}$.

The following two theorems are most useful when computing expected values of random variables.

THEOREM 1.2. (LINEARITY OF EXPECTATION). *Let* $X_1, \ldots, X_n$ *be random variables over* $(\Omega, \mathcal{F}, \mathbf{P})$, *then*

$$\mathbf{E}\left\{\sum_{i=1}^{n} c_i X_i\right\} = \sum_{i=1}^{n} c_i\, \mathbf{E}\{X_i\},$$

for any $c_1, \ldots, c_n \in \mathbb{R}$.

THEOREM 1.3. *Let Y be a random variable with finite expectation, X and Z vector-valued random variables. Then,*

- *There is a function g on $\mathbb{R}^k$ such that $\mathbf{E}\{Y|X\} = g(X)$ with probability one.*
- $\mathbf{E}\{Y\} = \mathbf{E}\{\mathbf{E}\{X|Y\}\}$.
- *If Y is a function of X then $\mathbf{E}\{Y|X\} = Y$.*

We will prove some results in the second part of the thesis about the convergence in probability of the running time of an algorithm. So, let us define this notion.

DEFINITION. (CONVERGENCE IN PROBABILITY). *Let $\{X_n\}_{n\in\mathbf{N}}$ be a sequence of random variables. We say that*

$$\lim_{n\to\infty} X_n = X, \quad \textit{in probability}$$

if for each $\varepsilon > 0$

$$\lim_{n\to\infty} \mathbf{P}\{|X_n - X| \geq \varepsilon\} = 0.$$

1.2 Inequalities

From now on, whenever a random variable X is given, a proper $(\Omega, \mathcal{F}, \mathbf{P})$ probability space is supposed to exist. We present several standard inequalities that we will use through out the thesis.

THEOREM 1.4. (CAUCHY-SCHWARZ INEQUALITY). *If the random variables X and Y have finite second moment (i.e., $\mathbf{E}\{X^2\} < \infty$ and $\mathbf{E}\{Y^2\} < \infty$), then*

$$|\mathbf{E}\{XY\}| \leq \sqrt{\mathbf{E}\{X^2\}\,\mathbf{E}\{Y^2\}}.$$

THEOREM 1.5. (HÖLDER'S INEQUALITY). *Let $p, q \in (1, \infty)$, such that $\frac{1}{p} + \frac{1}{q} = 1$. Let X and Y be random variables such that $\mathbf{E}\{|X^p|\} < \infty$, and*

$\mathbf{E}\{|Y^q|\} < \infty$. *Then.*

$$\mathbf{E}\{|XY|\} \leq (\mathbf{E}\{|X^p|\})^{1/p} (\mathbf{E}\{|Y^q|\})^{1/q}.$$

THEOREM 1.6. (MARKOV'S INEQUALITY). *Let X be a nonnegative random variable. Then for each $t > 0$,*

$$\mathbf{P}\{X \geq t\} \leq \frac{\mathbf{E}\{X\}}{t}.$$

THEOREM 1.7. (JENSEN'S INEQUALITY). *If f is a real-valued convex function on a finite or infinite interval, and X is a random variable with finite expectation, taking values in this interval, then*

$$f(\mathbf{E}\{X\}) \leq \mathbf{E}\{f(X)\}.$$

1.3 Graphs and trees

A *directed graph* G is a pair (V, E) where V is a finite set of *vertices* and E, the *edge* set, is a subset of $V \times V$. In an *undirected graph* we consider the edges unordered. The *degree* of a vertex $v \in V$ is the number of edges incident on it. A *walk* is a sequence $v_1, \ldots, v_t$ of vertices such that $(v_i, v_{i+1}) \in E$, for $i = 1, \ldots, t-1$. A *path* is a walk on which no vertices are repeated. We say that a graph G is connected if for every pair of vertices there is a path that connects them. A *cycle* is a walk such that its first and last points coincide. A *tree* T is a connected and acyclic graph. A *rooted* tree is a tree with a specially marked node, which we call *root* of T.

A rooted directed tree is called *m-ary* if every vertex has out-degree at most m and its children are numbered from 1 to m. A *binary tree* is a 2-ary

tree in which for every node each child is designated as a *left child* or *right child*. A *binary search tree* is a binary tree in which each vertex has an associated key coming from a totally ordered universe, such that all the keys associated to the nodes in its left subtree are smaller or equal than it and all the keys in its right subtree are greater than it.

1.4 Asymptotics

Many results in the thesis are given in asymptotic notation. We will now define these notions.

Let $f, g : \mathbb{N} \to \mathbb{R}^+$. Then,

$f = O(g)$ if and only if $\exists n_0 \in \mathbb{N}, c > 0 : f(n) \leq cg(n), \forall n \geq n_0$,

$f = \Omega(g)$ if and only if $\exists n_0 \in \mathbb{N}, c > 0 : f(n) \geq cg(n), \forall n \geq n_0$,

and

$$f = \Theta(g) \text{ if and only if } f = O(g) \text{ and } g = \Omega(g).$$

Furthermore,

$$f = o(g) \text{ if and only if } \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0,$$

$$f = \omega(g) \text{ if and only if } \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$$

PART I

Range Search

and

Nearest Neighbor

Algorithms

Chapter 2
Data Structures and Algorithms

In this chapter we define the problems we will study and the data structures and algorithms that we will be using to solve the proposed problems in the first part of the thesis. We introduce the squarish k-d trees as an alternative data structure to solve the problems at hand.

2.1 Range search and partial match

Range search is a fundamental problem in computational geometry and other related areas such as pattern recognition, statistics and database management systems. It is a problem commonly used within more complex problems. We can define formally the range search problem as follows:

THE RANGE SEARCH PROBLEM: Given are a set $S = \{u_1, \ldots, u_n\}$ of n points in $\mathbb{R}^k$ and $\mathcal{Q}$, a family of sets in $\mathbb{R}^k$. We wish to preprocess S into a data structure so that for a query range $Q \in \mathcal{Q}$, all the points in $Q \cap S$ can be reported efficiently.

There are several variants of this problem, for example: range counting (where we only need to report the number of points in $Q \cap S$), emptiness queries (we need only to decide whether $Q \cap S = \emptyset$) or extremal queries (where we report the set of points in $Q \cap S$ satisfying a specified extremal property). We should note now that the partial match query problem (report all points whose values match a given k-dimensional vector with possibly a number of wild-cards, e.g, we may search all points with values $(a_1, *, *, a_4, a_5, *)$ where $*$ denotes a wild card) is a particular instance of range search where the rectangles degenerate to products of points, intervals and real lines. For orthogonal range

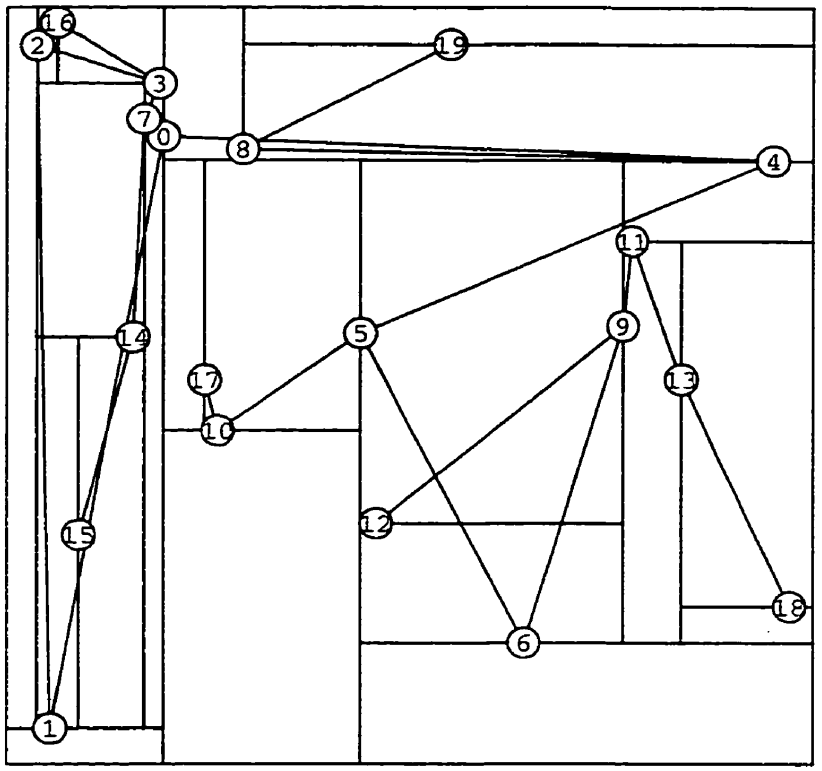

FIGURE 2.1. The k-d tree and its partition of $[0,1]^2$.

search, $\mathcal{Q}$ is the set of all rectangles. Different data structures have been used, depending of the definition of $\mathcal{Q}$, that help to solve the problem efficiently, such as quad trees, k-d trees, and range trees. For surveys see Bentley and Friedman (1979), Samet (1990a, 1990b), Agarwal (1997) and Matousek (1994). In a deterministic context, fractional cascading, a modification of the range tree technique, yields an $O(n(\log n)^{k-1})$ worst case space bound and $O((\log n)^{k-1}+N)$ worst case search time, for n data points, where N is the number of points in the query region. However, the implementation of this algorithm and data structure is quite cumbersome. The solution using k-d trees offers several advantages: it takes $O(kn)$ space for n data points, k-d trees are easily updated and maintained, it is simple to implement and comprehend, and it is useful for other operations besides orthogonal range search. Here we concentrate on the use of the k-d tree for solving orthogonal range search problems.

2.2 k-d trees

The k-d tree (Bentley, 1975) is a data structure used for storing multidimensional data. It is a binary tree in which each record contains k keys, right and left pointers to its subtrees, and an integer index between 1 and k that indicates which key in the record is used for splitting. On any path from the root, splitting is performed in a rotational fashion. For $k = 1$, we obtain the standard binary search tree. For any node u having index $j \in \{1, \ldots, k\}$, all nodes in its left subtree are such that their j^{th} key is less than the j^{th} key in u, and all nodes in its right subtree are such that their j^{th} key is greater or equal than the j^{th} key in u. The root is assigned index 1, and from there on the index of each node is determined by the depth of the node in the tree in a rotational fashion (see figure 2.1). The tree and partition shown in figure 2.1 are constructed by sequentially inserting the points. From now on the words rectangle and hyper-rectangle will be used interchangeably.

If we assume that the data belongs to $[0,1]^k$, then the insertion of $u_1, \ldots, u_n \in [0,1]^k$ in an initially empty k-d tree T creates a family of $2n+1$ rectangles that we call $\mathcal{R}_n$. We can associate with each data point u_{i+1} the hyper-rectangle in $[0,1]^k$ in the final partition generated by $u_1, \ldots, u_i$ in which it falls. Then each node (including external nodes) in T corresponds to a region of the unit hypercube. To fix ideas, for $1 \leq i \leq n$, we denote by $R_i \in \mathcal{R}_n$ the rectangle split by u_i. The $n+1$ leaf rectangles are also denoted R_i, with the index i now running from $n+1$ to $2n+1$. The set of these rectangles is denoted by $\mathcal{F}_n$. We will take the freedom of considering $\mathcal{R}_n$ and $\mathcal{F}_n$ as either the set of rectangles previously defined, or the set of indices of the respective rectangles. The dimensions of rectangle R_i are x_{ij}, for $1 \leq j \leq k$. In the two dimensional case the dimensions are denoted by x_i and y_i. It turns out that the shape of these rectangles is very important for the expected running time of partial match queries.

In order to solve range search, when k-d trees are used to store the data, we use a natural algorithm proposed by Bentley (1975) (see figure 2.2).

Range-Search(T, Q)

$u \leftarrow \text{root}[T], \Gamma \leftarrow \emptyset$

if $|T| = 0$ **then return** Γ

else if $u \in Q$ **then** $\Gamma \leftarrow \{u\}$

if $|T| = 1$ **then return** Γ

else $\ell \leftarrow \text{index}[u]$

case $u_\ell \leq z_\ell - m_\ell : \Gamma \leftarrow \Gamma \cup$ **Range-Search**$(T_{\text{right}(u)}, Q)$ (a)

$u_\ell \geq z_\ell + m_\ell : \Gamma \leftarrow \Gamma \cup$ **Range-Search**$(T_{\text{left}(u)}, Q)$ (b)

$z_\ell - m_\ell \leq u_\ell \leq z_\ell + m_\ell : \Gamma \leftarrow \Gamma \cup$**Range-Search**$(T_{\text{right}(u)}, Q)$
$\cup$**Range-Search**$(T_{\text{left}(u)}, Q)$ (c)

FIGURE 2.2. Bentley's range search algorithm.

Bentley's algorithm starts the search at the root. At each node, it looks at its index $j \in \{1, \ldots, k\}$, and compares the j^{th} key of the current node with the j^{th} range in the search region. If the range is entirely to the left, the search continues only on the left child of the node, if it is entirely to the right, then the search continues only at the right child. Otherwise, the search visits both subtrees.

The query time for orthogonal search depends upon many factors, such as the location of the query rectangle, and the distribution of the points. One may construct a median k-d tree off-line by splitting each time about the median, thus obtaining a perfectly balanced binary tree, in which ordinary point search takes $\Theta(\log n)$ worst-case time, and a partial match query with s coordinates specified takes worst-case time $O(n^{1-s/k} + N)$, where N is the number of points returned (see for example, Lee and Wong, 1977). Assuming the uniform model (the n data points independent and uniformly distributed random vectors on $[0, 1]^k$ and the specified entries in the query vector independent and uniform random variables), Flajolet and Puech (1986) computed the asymptotic expected running time of partial match query when s out of the k attributes are specified and k-d trees are used as data structures, and

found that if N_n denotes the time complexity of a partial match with s attributes specified, then

$$\mathbf{E}\{N_n\} = (1 + o(1))n^{1-s/k+\theta(s/k)}, \qquad (*)$$

where $0 \leq \theta(x) \leq 0.07$, for $x \in [0,1]$. In chapter 3 we give a new proof of their result, by means of probabilistic techniques, and show in chapter 4 how this result can be used to compute the expected running time of orthogonal range search. We also extend this result to convex sets in the plane.

We know much more about the expected time complexity of partial match queries when k-d trees are used as underlying data structures than the result of Flajolet and Puech (1986). Neininger (1999) showed that the first asymptotic term for $\mathbf{Var}\{N_n\}$ is $\Theta\left((\mathbf{E}\{N_n\})^2\right)$, and that

$$\frac{(N_n - \mathbf{E}\{N_n\})}{\sqrt{\mathbf{Var}\{N_n\}}}$$

tends in distribution to a non-degenerate limit law. That is, N_n is asymptotically not concentrated about $\mathbf{E}\{N_n\}$. Their method of proof uses contractions, and may also be used to analyze partial match queries for random quad trees (Neininger and Rüschendorf 1999), thus extending results of Flajolet, Gonnet, Puech and Robson (1990, 1992).

Partial match queries have also been analyzed for locally balanced kdt trees, a balanced version of random k-d trees, by Cunto, Lau and Flajolet (1989). A kdt tree is a k-d tree where each subtree of size greater than $2t$ has at least t nodes on each of its subtrees. Note that in particular if a subtree has $2t + 1$ nodes, then the key with the median of the component which is being used as discriminatory in the whole subtree at this level is stored in its root. Cunto, et al. (1989) proved that for kdt trees $\mathbf{E}\{N_n\} = \Theta\left(n^{1-s/k+\hat{\theta}(s/k,t)}\right)$, where $\hat{\theta}(x,t) \to 0$, as $t \to \infty$. So, kdt trees improve with respect to the behavior of k-d trees as t grows.

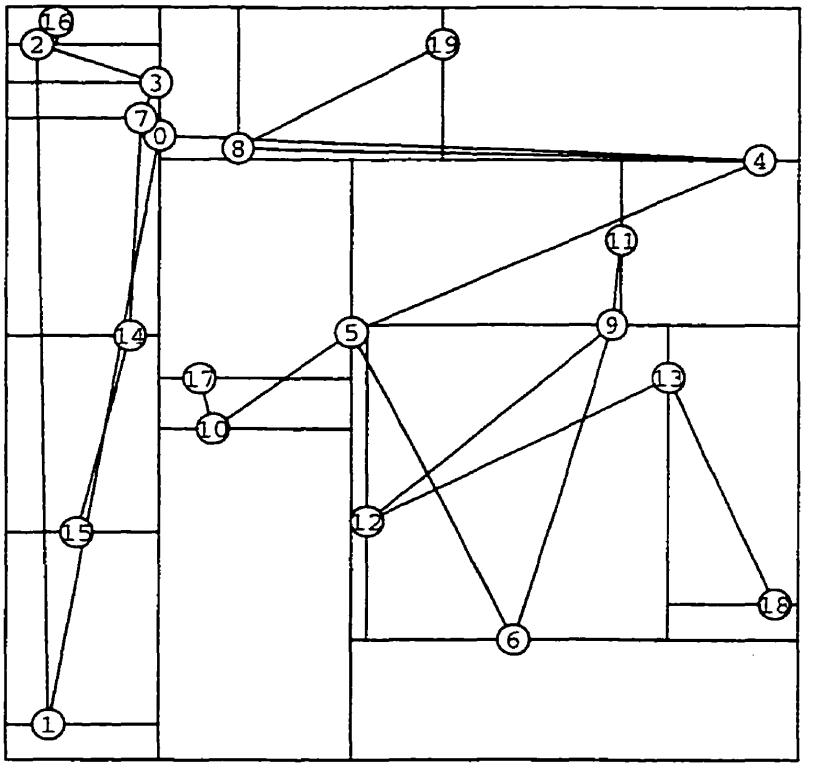

FIGURE 2.3. The squarish k-d tree and its partition of the plane

We should note that in the previous types of k-d trees, deletion operations are not easy to implement, as it may involve a tedious reorganization of the whole tree. Furthermore the distribution of the tree may be changed by the insertion and deletion of nodes. To overcome these problems Duch, Estivill-Castro and Martinez (1998), and Martinez, Panholzer and Prodinger (1998) proposed the relaxed k-d tree. For this tree, instead of rotating cyclically the discriminatory coordinate, we cut directions uniformly at random. Duch, et al. (1998) and Martínez, et al. (1998) showed that for the relaxed k-d trees $\mathbf{E}\{N_n\} = \Theta\left(n^{\alpha-1}\right)$, where $\alpha = 1/2(1+\sqrt{9-8s/k})$, which is worse than for ordinary random k-d trees.

2.3 Squarish k-d trees

Squarish k-d trees are a minor modification of k-d trees, first proposed in

this thesis. The insertion procedure is modified so that each time a rectangle is split by a newly inserted leaf point, the longest side of its rectangle is cut, that is, the cut is a $(k-1)$-dimensional hyper-plane through the new point perpendicular to the longest edge of the rectangle. Figure 2.3 shows the squarish tree and its associated partition for the same set of points in figure 2.1. In case that there is more than one longest edge, we toss a perfect coin to decide on one of the longest edges for cutting. It is natural to conjecture that if the data points are taken in the unit hypercube, the squarish partition rule should create a more squarish looking rectangle partition than if we use a k-d tree partition criterion. This will be apparent from our results.

In chapters 3 and 4 we show that the elongated nature of the rectangles generated by the standard k-d tree partition rule explains the poor performance of random k-d trees with respect to partial match and range search queries. It is precisely because of the more squarish nature of the rectangles generated by the squarish partition rule that they have a better performance on the average with respect to partial match queries. This seemingly small change makes the expected time behavior of lower-dimensional partial match queries behave as for perfectly balanced complete k-d trees on n nodes, namely $\Theta(n^{1-s/k})$. This is in contrast with the k-d trees reviewed in the previous section, where it was seen that none of them are optimal on the average with respect to partial match queries.

2.4 Nearest neighbor search

In chapter 5 we analyze two algorithms to solve the nearest neighbor problem. In algorithm **A** in dimension k, we perform range search queries with square boxes of side length $k^{t/2}/n^{1/k}$, for $t = 0, 1, 2, \ldots$, until $T^* + 1$, where T^* is the first non-empty box. An important quantity that we define is

$$\rho_k = \max_{0 \le s \le k} \theta(s/k),$$

where $\theta(\cdot)$ is the function appearing in $(*)$. If the data structure we use is a k-d tree, and if T_t is the time complexity of range search on a square

box of side length $k^{t/2}/n^{1/k}$, by applying the results in chapter 4 about the complexity of range search queries on k-d trees we prove that

$$\mathbf{E}\{T_t\} \leq C\left(k^{\frac{(k-1)t}{2}} n^{\rho_k} + k^{\frac{kt}{2}}\right) .$$

for some $C > 0$ not depending upon t or n. Using the previous observation, we will prove that the expected complexity time of algorithm **A**, in dimension k, is $\Theta(n^{\rho_k})$. When the data structure we use is a squarish k-d tree, then we prove that the expected time complexity of algorithm **A** is $O(\log n \log\log n)$.

In algorithm **B**, we insert the query point Z in the k-d tree containing the data and perform a range search centered at Z with dimensions twice the distance of Z to its parent in the k-d tree. The nearest neighbor is reported among all points returned by the orthogonal range search. We analyze, in dimension 2, algorithm **B** on squarish k-d trees and show that its expected complexity is $O(\log^2 n)$.

2.5 The model

We consider n independent and uniformly distributed points $U_1, \ldots, U_n$ on $[0,1]^k$. We store them in order in a k-d tree and call the actual data $u_1, \ldots, u_n$. The query rectangle Q is $Z + [-m_1, m_1] \times \cdots \times [-m_k, m_k]$, where $m_j \geq 0$ for all j, where the m_j's are fixed (that is, they may depend upon n only) and Z is uniformly distributed on $[0,1]^k$ and independent of all $U_1, \ldots, U_n$. We denote by $\Delta_j = 2m_j$, for $1 \leq j \leq k$. Note that partial match queries are range search queries such that for some $S \subseteq \{1, \ldots, k\}$, for each $j \in S$, $m_j = 0$, and for each $j \notin S$, $m_j = 1$. We call this the *uniform model*. Note that the definitions made for the deterministic case in section 2.2 can be immediately extended to the uniform model.

When each component of the data comes from a continuous distribution, the sequence of ranks in each point forms a random permutation. The distribution of the k-d trees evolving from n points coincides with the distribution of binary trees evolving from the successive insertion of n 1-dimensional data

points from a common distribution. The distributions and moments of associated random variables are exactly the same as their analogues in binary trees.

2.6 Partial match and range search queries

Because for each node in the tree T there is associated a region in $[0,1]^k$, a node u in T is visited by Bentley's range search algorithm if and only if the query rectangle Q intersects its associated rectangle. A leaf rectangle is visited if its associated rectangle R_i intersects Q. Let N_n be the time complexity of Bentley's orthogonal range search, then by the previous

$$N_n = \sum_{i=1}^{2n+1} 1_{[R_i \cap Q \neq \emptyset]}.$$

For $1 \leq s \leq k$, and $v_{j_1}, \ldots, v_{j_s} \in [0,1]$, partial match query asks for all points in $\{u_1, \ldots, u_n\}$ satisfying $u_{ij_t} = v_{j_t}$, for all $1 \leq t \leq s$. We say that the query fixes coordinates $j_1, \ldots, j_s$. We also define L as the set all of points in $[0,1]^k$ whose j_t-th coordinate is equal to v_{j_t}, for all $1 \leq t \leq s$. In a partial match query, we let the s fixed coordinates be independent and uniformly distributed over $[0,1]$, and the n data points be random independent uniformly distributed vectors on $[0,1]^k$.

We first relate the expected time complexity of partial match to that of range search. The following proposition allows us to compute the expected time complexity of range search by using results about partial match queries.

PROPOSITION 2.1 *Given is a random k-d tree based on i.i.d. random variables $U_1, \ldots, U_n$, uniformly distributed on $[0,1]^k$. Consider a random partial match query, in which $s > 0$ of the k fields are specified. Let N_n be the number of comparisons that Bentley's orthogonal range search performs. Let S be the set of specified coordinates. Then*

$$\mathbf{E}\{N_n\} = \mathbf{E}\left\{\sum_{i=1}^{2n+1} \prod_{j \in S} X_{ij}\right\}.$$

where $X_{ij}, 1 \leq j \leq k$, is the length of the side along the j-th coordinate of rectangle R_i in $\mathcal{R}_n$.

PROOF. Note that $\mathbf{P}\{L \cap R_i \neq \emptyset \,|\, U_1, \ldots, U_n\} = \prod_{j \in S} X_{ij}$. Thus we have,

$$\mathbf{E}\{N_n\} = \mathbf{E}\left\{\sum_{i=1}^{2n+1} 1_{[L \cap R_i \neq \emptyset]}\right\}$$
$$= \sum_{i=1}^{2n+1} \mathbf{P}\{L \cap R_i \neq \emptyset\} = \mathbf{E}\left\{\sum_{i=1}^{2n+1} \prod_{j \in S} X_{ij}\right\}. \square$$

Chapter 3
Partial Match

In this chapter we present the probabilistic analysis for partial match queries using the uniform model, when the underlying data structure used is either a k-d tree or a squarish k-d tree. We present a new probabilistic proof of a result from Flajolet and Puech (1986) about the expected complexity time of partial match when using k-d trees. We also give the first analysis of the expected complexity time of partial match for squarish k-d trees.

3.1 The results of Flajolet and Puech

In a random vertical partial match query on a 2-d tree, we take a uniformly distributed value $Z \in [0,1]$, and visit all nodes in the tree whose rectangle cuts the vertical line at Z. The probability of hitting a rectangle with dimensions $X_i \times Y_i$ is of course X_i, so that the expected number of nodes visited, and hence, the expected time for a partial match query, is simply $\mathbf{E}\left\{\sum_{i=1}^{2n+1} X_i\right\}$, where the sum is taken over all $2n+1$ rectangles in the partition. A similar formula holds of course for horizontal partial match queries. The previous idea clearly generalizes to arbitrary dimensions. As we saw in the previous chapter, in dimension k, if $S \subseteq \{1 \ldots, k\}$, with $|S| = s$, is the set of fixed attributes and N_n denotes the number of comparisons that Bentley's orthogonal range search performs when fixing the coordinates in S, then we have that

$$\mathbf{E}\{N_n\} = \mathbf{E}\left\{\sum_{i=1}^{2n+1} \prod_{j \in S} X_{ij}\right\}.$$

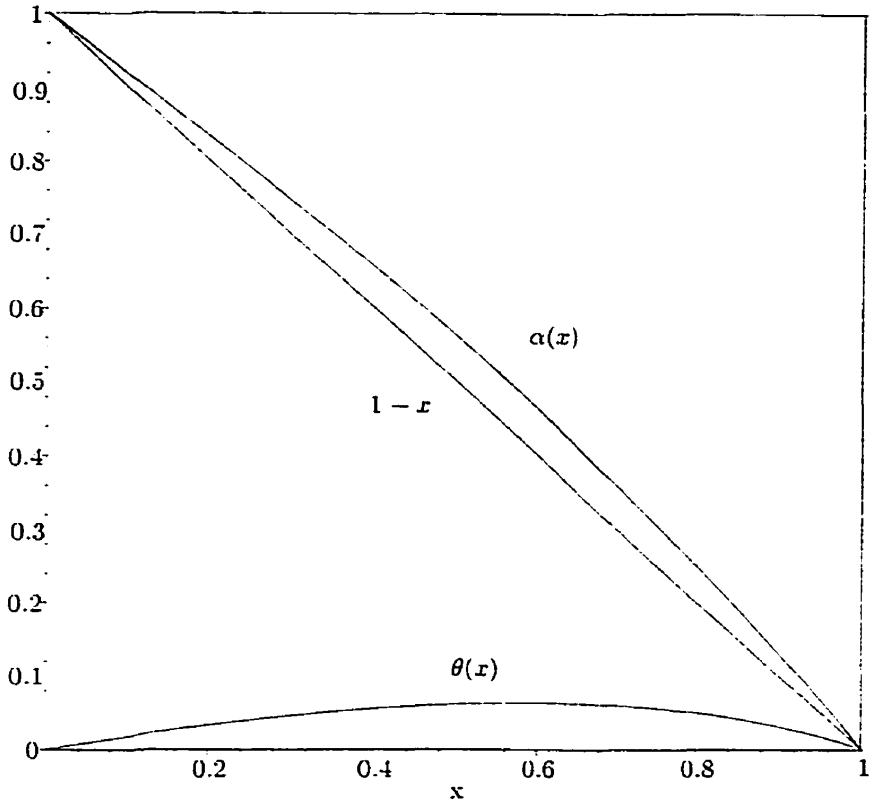

FIGURE 3.1. The top curve is the Flajolet-Puech function $\alpha(\cdot)$. The bottom curve is the function $\theta(\cdot)$.

THEOREM 3.1. (FLAJOLET AND PUECH (1986)). *Given is a random k-d tree based on i.i.d. random variables $U_1, \ldots, U_n$, distributed uniformly on $[0,1]^k$. Consider a random partial match query, in which s of the k fields are specified with $k > s \geq 0$. Let N_n be the number of comparisons that Bentley's orthogonal range search performs. Then*

$$\mathbf{E}\{N_n\} = (c + o(1))n^{\alpha(s/k)},$$

where c is a constant depending on the indices of the s fixed coordinates and for $0 \leq u \leq 1$, $\alpha(u) = 1 - u - \theta(u)$ where $\theta(u)$ is the root $\theta \in [0,1]$ of the equation $(\theta + 3 - x)^x(\theta + 2 - x)^{1-x} - 2 = 0$.

We call the function $\alpha(\cdot)$ in the previous theorem. the Flajolet-Puech $\alpha(\cdot)$ function. An alternative formulation for the previously defined function

is as follows:

$$\alpha(u) = \max_{0 \le t \le 1} \left\{ t + 2 \left(\frac{1-t}{1-u} \right)^{1-u} \left(\frac{t}{u} \right)^{u} - 2 \right\}.$$

Note in particular that α is decreasing, and that $1 - u \le \alpha(u) \le 1.07 - u$. Particular values of interest are $\alpha(0) = 1$, $\alpha(1/2) = 0.5616\ldots$, $\alpha(1/3) = 0.7162\ldots$, $\alpha(2/3) = 0.3949\ldots$, $\alpha(1) = 0$ (see figure 3.1). As we commented in the previous chapter, much more is known about the complexity of partial match queries. Neininger (1999) proved that if s out of the k fields are fixed, then for the time complexity of random partial match we have that.

$$\frac{N_n - \mathbf{E}\{N_n\}}{n^{\alpha(s/k)}} \xrightarrow{\ell_2} X,$$

where X is a non-degenerate random variable, and ℓ_2 is defined as follows:

$$\ell_2(\mu, \nu) = \inf\{\|X - Y\|_2 ; X \stackrel{\mathcal{L}}{=} \mu, Y \stackrel{\mathcal{L}}{=} \nu\}.$$

Furthermore, he proves that $\mathbf{Var}\{N_n\} = (c + o(1))(\mathbf{E}\{N_n\})^2$, where the constant c depends on the fixed coordinates. In the next section, we merely offer an alternative entirely probabilistic proof of one half of theorem 3.1.

3.2 Probabilistic proof for the theorem of Flajolet and Puech

The arguments of our proof can be traced back to Devroye (1986).

THEOREM 3.2. *For fixed s with $0 < s < k$, there exist constants C and C' depending upon s and k only such that, for all subsets $S \subseteq \{1, \ldots, k\}$ with $|S| = s$,*

$$C' n^{\alpha(s/k)} \le \mathbf{E}\left\{ \sum_{i=1}^{2n+1} \prod_{j \in S} X_{ij} \right\} \le C n^{\alpha(s/k)}.$$

PROOF. We prove the upper bound only. The proof uses an embedding argument that constructs an equivalent k-d tree. Assume without loss of

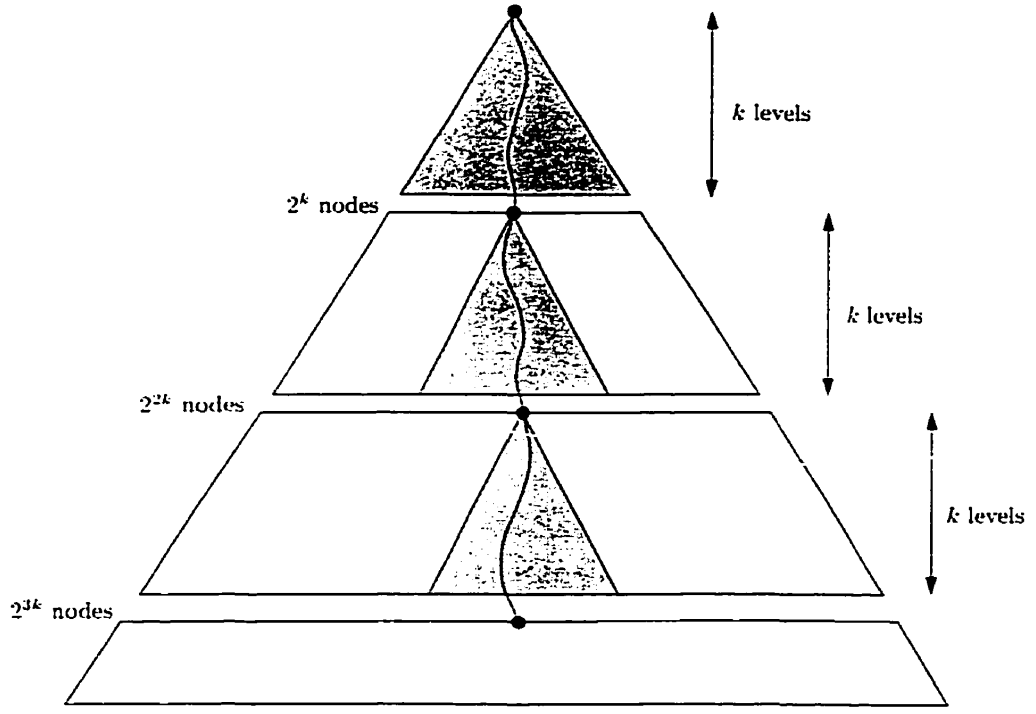

FIGURE 3.2. Tree showing argument in proof of theorem 3.2.

generality that the set S consists of the first s coordinates in the rotation (the other cases are not equivalent, but trivially similar). A split along coordinate j will be called a j-split. To determine a split, we just need a uniform $[0,1]$ random variable. So, the construction of the k-d tree may be viewed recursively as follows: at the root, the root rectangle $R_1 = [0,1]^k$ is subjected to a 1-split based on a uniform $[0,1]$ random variable U. One data point is associated with the root (this requires $k-1$ other uniformly distributed coordinates, but they will not be needed for what we need to study), and the sizes of the subtrees associated with the two sub-rectangles are multinomially distributed with parameters $(n-1, U, 1-U)$. We may apply this procedure recursively, rotating of course the axes about which we split. After k rounds, thus for rectangles at distance k from the root, the dimensions of a rectangle are described by a vector $(V_1 \ldots, V_k)$, with independent uniform $[0,1]$ components. As a binomial(N,p), where N is binomial(n,q), is binomial(n,pq), we see that the size of the subtree associated with the rectangle with dimensions

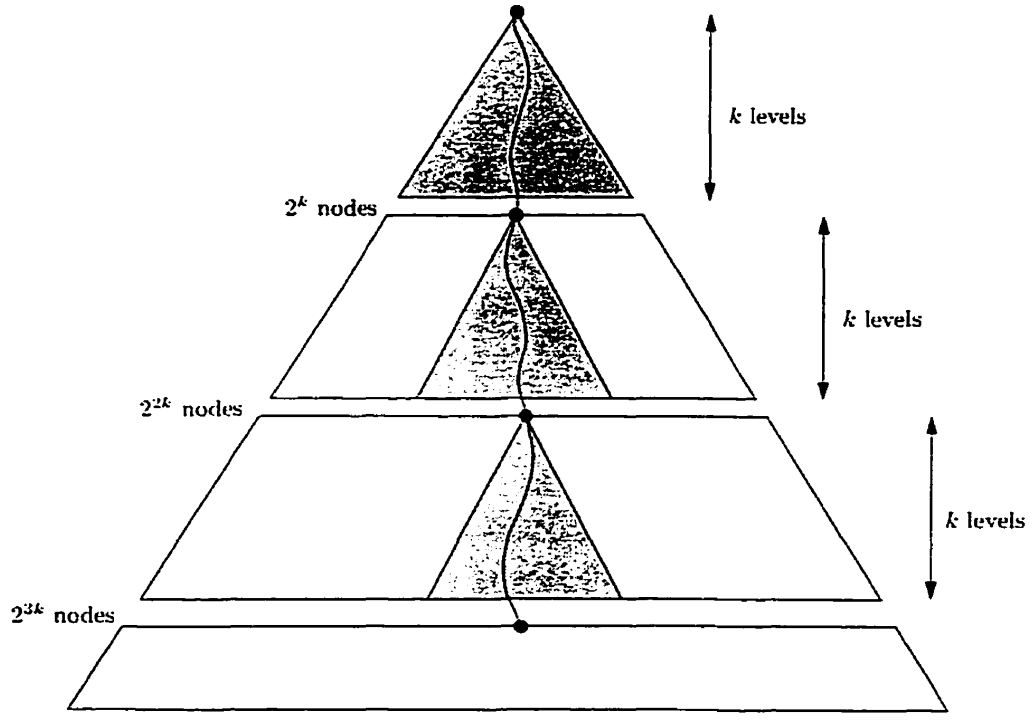

FIGURE 3.2. Tree showing argument in proof of theorem 3.2.

generality that the set S consists of the first s coordinates in the rotation (the other cases are not equivalent, but trivially similar). A split along coordinate j will be called a j-split. To determine a split, we just need a uniform $[0,1]$ random variable. So, the construction of the k-d tree may be viewed recursively as follows: at the root, the root rectangle $R_1 = [0,1]^k$ is subjected to a 1-split based on a uniform $[0,1]$ random variable U. One data point is associated with the root (this requires $k-1$ other uniformly distributed coordinates, but they will not be needed for what we need to study), and the sizes of the subtrees associated with the two sub-rectangles are multinomially distributed with parameters $(n-1, U, 1-U)$. We may apply this procedure recursively, rotating of course the axes about which we split. After k rounds, thus for rectangles at distance k from the root, the dimensions of a rectangle are described by a vector $(V_1, \ldots, V_k)$, with independent uniform $[0,1]$ components. As a binomial(N,p), where N is binomial(n,q), is binomial(n,pq), we see that the size of the subtree associated with the rectangle with dimensions

$(V_1, \ldots, V_k)$ is stochastically not larger than a binomial$(n, \prod_{i=1}^{k} V_i)$ random variable N. If $N = 0$, then the rectangle is either non-existent or a leaf in the final k-d tree. With this mechanism, our tree is an infinite complete binary tree. The actual k-d tree with $2n+1$ rectangles is a subtree of the tree whose nodes represent rectangles R such that $N = \text{binomial}(n, |R|) > 0$. These N's are dependent, but that will not matter in what follows, by linearity of expectation. We note thus that with each node in the infinite tree, an independent uniform $[0, 1]$ random variable is associated, and that the size of a rectangle R whose path from the root to the parent of the rectangle node has uniform random variables $V_1, V_2, \ldots$ is given by

$$(V_1 V_{k+1} V_{2k+1} \cdots, V_2 V_{k+2} V_{2k+2} \cdots, \ldots, V_k V_{2k} V_{3k} \cdots) .$$

Returning to the problem at hand, we introduce $V(R)$ and $W(R)$ for a rectangle R at distance ℓ from the root. Here $V(R)$ is the product of all uniforms on that path to the root that correspond to j-splits, $1 \le j \le s$, and $W(R)$ is the product for $s+1 \le j \le k$. Clearly, $|R| = V(R)W(R)$. The quantity of interest to us is

$$\mathbf{E}\left\{\sum_{i=1}^{2n+1} V(R_i)\right\} \le$$

$$2\sum_{\ell=0}^{\infty} \mathbf{E}\left\{\sum_{\text{all rectangles } R \text{ at depth } \ell} V(R) 1_{[\text{binomial}(n,|R|)>0]}\right\} .$$

Here we consider of course the infinite tree. Leaf nodes in the k-d tree have of course zero cardinality, but their parents do not. For this reason, we consider only parent nodes, which explains the coefficient 2. Let Z_r and Z'_m represent independent products of r and m independent uniform $[0, 1]$ random variables respectively. Then, by looking at levels that are multiples of k only, the last upper bound is not more than $2^{k+1} + 2^{k+1}M$, where

$$M = \sum_{\ell=1}^{\infty} 2^{k\ell} \mathbf{E}\left\{Z_{s\ell} 1_{\left[\text{binomial}(n, Z_{s\ell} Z'_{(k-s)\ell})>0\right]}\right\} .$$

To study M, note first that a uniform $[0,1]$ random variable is distributed as e^{-E}, where E is a standard exponential random variable. Thus, $Z_{s\ell}$ is distributed as $e^{-G_{s\ell}}$, where G_r denotes a gamma(r) random variable, that is, a random variable with density

$$f(x) = \frac{x^{r-1}e^{-x}}{\Gamma(r)} \ , \ x > 0 \ .$$

Similarly, $Z'_{(k-s)\ell}$ is distributed as $e^{-G_{(k-s)\ell}}$. We write from now on G and G' for independent gamma random variables. We have then.

$$\begin{aligned}
M &\leq \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ Z_{s\ell} \min\left(1, nZ_{s\ell}Z'_{(k-s)\ell}\right)\right\} \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell} \left(\mathbf{E}\left\{ Z_{s\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}\geq 1\right]}\right\} + \mathbf{E}\left\{ nZ_{s\ell}^2 Z'_{(k-s)\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}< 1\right]}\right\}\right) \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ Z_{s\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}\geq 1\right]}\right\} + \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ nZ_{s\ell}^2 Z'_{(k-s)\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}< 1\right]}\right\} \\
&= I + II \ .
\end{aligned}$$

First we handle I. We have

$$\begin{aligned}
I &= \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ Z_{s\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}\geq 1\right]}\right\} \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ e^{-G_{s\ell}} 1_{\left[G_{s\ell}+G'_{(k-s)\ell}\leq \log n\right]}\right\} \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell} \int_{\substack{x+y<\log n \\ x\geq 0, y\geq 0}} e^{-x}\, \frac{x^{s\ell-1}y^{(k-s)\ell-1}}{\Gamma(s\ell)\Gamma((k-s)\ell)}\, e^{-x-y}\, dx\, dy \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell} \int_{0<z<\log n} \int_{0<t<1} z^{k\ell-1} e^{-2tz-(1-t)z}\, \frac{t^{s\ell-1}(1-t)^{(k-s)\ell-1}}{\Gamma(s\ell)\Gamma((k-s)\ell)}\, dt\, dz \\
&\qquad \text{(by the transform } x = tz, y = (1-t)z, 0 < t < 1) \ .
\end{aligned}$$

Similarly, II yields

$$\begin{aligned}
II &= \sum_{\ell=1}^{\infty} 2^{k\ell}\, \mathbf{E}\left\{ nZ_{s\ell}^2 Z'_{(k-s)\ell} 1_{\left[nZ_{s\ell}Z'_{(k-s)\ell}< 1\right]}\right\} \\
&= \sum_{\ell=1}^{\infty} 2^{k\ell} n \int_{\log n\leq z} \int_{0<t<1} z^{k\ell-1} e^{-3tz-2(1-t)z}\, \frac{t^{s\ell-1}(1-t)^{(k-s)\ell-1}}{\Gamma(s\ell)\Gamma((k-s)\ell)}\, dt\, dz
\end{aligned}$$

so that

$$I + II =$$

$$\sum_{\ell=1}^{\infty} 2^{k\ell} \int_{0<z<\infty} \int_{0<t<1} z^{k\ell-1} \min(1, ne^{-z})\, e^{-2tz-(1-t)z}\, \frac{t^{s\ell}(1-t)^{(k-s)\ell}}{t(1-t)\Gamma(s\ell)\Gamma((k-s)\ell)}\, dt\, dz$$

We first estimate the sum over ℓ, taking only those terms that depend upon ℓ:

$$III = \sum_{\ell=1}^{\infty} \frac{a^{\ell}}{\Gamma(s\ell)\Gamma((k-s)\ell)} ,$$

where $a = 2^k x^s y^{(k-s)}$, and we recall that $x = tz, y = (1-t)z$. Thus,

$$I + II = \int_{0<z<\infty} \int_{0<t<1} III \times \frac{\min(1, ne^{-z})\, e^{-2tz-(1-t)z}}{zt(1-t)}\, dt\, dz .$$

Employing the Stirling approximation

$$\Gamma(\ell) = \left(\frac{\ell}{e}\right)^{\ell} \sqrt{\frac{2\pi}{\ell}} e^{\frac{\theta}{12\ell}}$$

for some $\theta \in [0,1]$ (Whittaker and Watson, 1927, p. 253), we have for $\ell > 0$,

$$\frac{\Gamma(s\ell)\Gamma((k-s)\ell)}{\Gamma(k\ell)} \geq \sqrt{2\pi} e^{-1/12} \sqrt{\frac{k}{s(k-s)\ell}} \left(\frac{s^s (k-s)^{k-s}}{k^k}\right)^{\ell} .$$

Defining $u = s/k$, and

$$\beta = \frac{2x^u y^{1-u}}{u^u (1-u)^{1-u}} = 2z \left(\frac{t}{u}\right)^u \left(\frac{1-t}{1-u}\right)^{1-u} ,$$

we obtain the bound

$$\begin{aligned} III &\leq \frac{e^{1/12}\sqrt{s(k-s)}}{\sqrt{2\pi k}} \sum_{\ell=1}^{\infty} \sqrt{\ell} \frac{\left(\frac{a^{1/k}}{u^u(1-u)^{1-u}}\right)^{k\ell}}{\Gamma(k\ell)} \\ &= \frac{e^{1/12}\sqrt{s(k-s)}}{\sqrt{2\pi k}} \sum_{\ell=1}^{\infty} \frac{\sqrt{\ell}\beta^{k\ell}}{\Gamma(k\ell)} \\ &= \frac{e^{1/12}\sqrt{u(1-u)}\, e^{\beta}}{\sqrt{2\pi}} \sum_{\ell=1}^{\infty} \frac{(k\ell)^{\frac{3}{2}} \beta^{k\ell} e^{-\beta}}{(k\ell)!} . \end{aligned}$$

We now show that there is a constant $C_0 > 0$ such that for all $\beta > 0$,

$$\frac{e^{1/12} e^{\beta}}{\sqrt{2\pi}} \sum_{\ell=1}^{\infty} \frac{(k\ell)^{\frac{3}{2}} \beta^{k\ell} e^{-\beta}}{(k\ell)!} \leq C_0 e^{\beta} \beta^{3/2},$$

and thus,

$$III \le C_0 \sqrt{u(1-u)}\, e^{\beta} \beta^{\frac{3}{2}}.$$

For $\beta > 1$, we have by Jensen's inequality

$$\begin{aligned}\frac{e^{1/12} e^{\beta}}{\sqrt{2\pi}} \sum_{\ell=1}^{\infty} \frac{(k\ell)^{\frac{3}{2}} \beta^{k\ell} e^{-\beta}}{(k\ell)!} &\le \frac{e^{1/12} e^{\beta}}{\sqrt{2\pi}} \mathbf{E}\left\{\text{Poisson}^2(\beta)\right\}^{3/4} \\ &\le \frac{e^{1/12} e^{\beta}}{\sqrt{2\pi}} \left(\beta^2 + \beta\right)^{3/4} \le \frac{e^{1/12} 2^{3/4}}{\sqrt{2\pi}} e^{\beta} \beta^{3/2}.\end{aligned}$$

For $\beta \le 1$,

$$\frac{e^{1/12} e^{\beta}}{\sqrt{2\pi}} \sum_{\ell=1}^{\infty} \frac{(k\ell)^{\frac{3}{2}} \beta^{k\ell} e^{-\beta}}{(k\ell)!} \le \frac{e^{1/12} e^{\beta} \beta^{k}}{\sqrt{2\pi}} \sum_{j=1}^{\infty} \frac{j^{3/2}}{j!} \le C^* e^{\beta} \beta^{3/2},$$

as $\sum_{j=1}^{\infty} \frac{j^{3/2}}{j!}$ converges and $k \ge 2$. Resubstitution yields

$$\begin{aligned}&I + II \\ &\le C_0 \int_0^{\infty} \int_0^1 \sqrt{\frac{zu(1-u)\left(\frac{t}{u}\right)^{3u}\left(\frac{1-t}{1-u}\right)^{3(1-u)}}{t^2(1-t)^2}} \min(1, ne^{-z})\, e^{z\left(2\left(\frac{t}{u}\right)^u \left(\frac{1-t}{1-u}\right)^{1-u} - t - 1\right)}\, dt\, dz \\ &= C \int_0^{\infty} \sqrt{z} \min(1, ne^{-z}) \left[\int_0^1 h(t) e^{zg(t)}\, dt\right] dz\ ,\end{aligned}$$

where $C = C_0 \sqrt{u(1-u)/(u^{3u}(1-u)^{3(1-u)})}$, $h(t) = t^{3u/2-1}(1-t)^{3(1-u)/2-1}$, and

$$g(t) = 2\left(\frac{t}{u}\right)^u \left(\frac{1-t}{1-u}\right)^{1-u} - t - 1\ .$$

The behavior of g is easily established: by definition of the Flajolet-Puech function, we have $\sup_{0<t<1} g(t) = \alpha(u)$, and the maximum occurs at $t_0 \in (0,1)$. Furthermore, g is unimodal and locally concave about t_0. Hence, there exists a constant $\nu > 0$ such that $g(t) \le \alpha(u) - \nu(t - t_0)^2$ for all $t \in (0,1)$. Pick $\epsilon > 0$ such that $B = (t_0 - \epsilon, t_0 + \epsilon) \subseteq (0,1)$. Then

$$\begin{aligned}&\int_{0<t<1} h(t) e^{zg(t)} dt \\ &\qquad \le \sup_B h(t) \int_{-\infty}^{\infty} e^{z(\alpha(u) - \nu(t-t_0)^2)} dt + e^{z(\alpha(u) - \nu\epsilon^2)} \int_0^1 h(t) dt \\ &\qquad \le \frac{D}{\sqrt{z}} e^{z\alpha(u)} + D' e^{z(\alpha(u) - \nu\epsilon^2)}\end{aligned}$$

where D and D' are positive constants only depending upon u (through the function h and the constant ν). Resubstitution now yields

$$I + II \leq C \int_0^\infty \sqrt{z} \min(1, ne^{-z}) \left(\frac{D}{\sqrt{z}} e^{z\alpha(u)} + D' e^{z(\alpha(u) - \nu\epsilon^2)} \right) dz \ .$$

Split the integral over $(0, \log n)$ and $(\log n, \infty)$, and verify that the result is $O(n^{\alpha(u)})$, and that all multiplicative constants indeed only depend upon u and k. □

3.3 Random partial match queries with squarish 2-d trees

In this section, we prove that a random partial match query in a random squarish 2-d tree takes expected time $\Theta(\sqrt{n})$ as opposed to $\Theta(n^{0.5616\ldots})$ for random 2-d trees (see theorem 3.3). We start with the following observation, that immediately follows by considering the random growth of our k-d trees. Of course, it implies that the joint distribution of the ordered volumes of the $n + 1$ leaf rectangles is identical for both random k-d trees considered here.

LEMMA 3.1 *Consider a random k-d tree or a random squarish k-d tree. Then, the volumes of the rectangles in $\mathcal{F}_n$ are distributed as the set $\mathcal{V}_n$ of the consecutive spacings between the order statistics of n i.i.d. random variables, uniformly distributed on* $[0, 1]$.

The next result will be useful when we compute the expected time complexity of range search.

LEMMA 3.2 *Consider a random k-d tree or squarish k-d tree constructed from $U_1, \ldots, U_n$ independent uniformly distributed random variables over $[0, 1]^k$. Let $\mathcal{R}_n$ be the rectangles in the partition defined by either the random k-d tree or the random squarish k-d tree based on $U_1, \ldots, U_n$. Let X_{ij} be the length on the j^{th} coordinate of the i^{th} rectangle. Then,*

$$\mathbf{E}\left\{ \sum_{i=1}^{2n+1} X_{i1} \cdots X_{ik} \right\} = 2H_{n+1} - 1,$$

where H_n is the n^{th} harmonic number.

PROOF. First, note that for any $1 \leq i \leq n$, $X_{i1} \cdots X_{ik}$ is the volume $|R_i|$ of the rectangle R_i. Note that if $U_1, \ldots, U_i$ have already been inserted in $[0,1]^k$, and U_{i+1} is a new point, then the size of the two rectangles generated by U_{i+1} is equal to the size of the rectangle in the final partition of $[0,1]^k$ in which U_{i+1} falls. Let us denote by $R(U_{i+1})$ this rectangle. Thus,

$$\mathbf{E}\left\{\sum_{i=1}^{2n+1} X_{i1} \cdots X_{ik}\right\} = 1 + \sum_{i=0}^{n-1} \mathbf{E}\left\{\mathbf{E}\left\{|R(U_{i+1})| \mid U_1, \ldots, U_i\right\}\right\},$$

where the 1 accounts for the root rectangle. We claim that $\mathbf{E}\left\{|R(U_{i+1})|\right\} = \frac{2}{i+2}$. Note that the claim is obviously true for $i = 0$. Now, suppose that $U_1, \ldots, U_i$ have already been inserted in the tree, so that there are $i+1$ external nodes. These external nodes represent the $i+1$ rectangles partitioning $[0,1]^k$. Let these rectangles be $S_1, \ldots, S_{i+1}$, and let the numbering be so that the leaves are taken from left to right, in order of appearance as leaves in the k-d tree of $U_1, \ldots, U_i$. Then,

$$\begin{aligned}
\mathbf{E}\left\{|R(U_{i+1})|\right\} &= \mathbf{E}\left\{\mathbf{E}\left\{\sum_{\ell=1}^{i+1} \mathbf{1}_{[U_{i+1} \in S_\ell]} |S_\ell| \mid U_1, \ldots, U_i\right\}\right\} \\
&= \mathbf{E}\left\{\sum_{\ell=1}^{i+1} |S_\ell| \, \mathbf{P}\left\{U_{i+1} \in S_\ell \mid U_1, \ldots, U_i\right\}\right\} \\
&= \mathbf{E}\left\{\sum_{\ell=1}^{i+1} |S_\ell|^2\right\}.
\end{aligned}$$

It is well known that $(|S_1|, \ldots, |S_{i+1}|)$ are jointly distributed as uniform spacings, that is the lengths of the intervals on $[0,1]$ defined by an i.i.d. uniform $[0,1]$ sample of size i. All these spacings are identically distributed following a beta$(1,i)$ distribution. If B is a beta$(1,i)$ random variable, then we have $\mathbf{E}\{B\} = 1/(i+1)$ and $\mathbf{E}\{B^2\} = 2/((i+1)(i+2))$. Therefore,

$$\mathbf{E}\left\{|R(U_{i+1})|\right\} = (i+1)\,\mathbf{E}\{B^2\} = \frac{2}{i+2}.$$

and thus,

$$1 + \sum_{i=0}^{n-1} \mathbf{E}\left\{|R(U_{i+1})|\right\} = 1 + 2(H_{n+1} - 1). \quad \square$$

We state now the theorem that shows the behavior of partial match when using squarish 2-d trees.

THEOREM 3.3. *For a random squarish 2-d tree,*

$$\frac{\sqrt{\pi n}}{3} \leq \mathbf{E}\left\{\sum_{i=1}^{2n+1} Y_i\right\} \leq 180\sqrt{n}.$$

The same result holds for $\mathbf{E}\left\{\sum_{i=1}^{2n+1} X_i\right\}$. *Hence, the expected time for a random partial match query is* $\Theta(\sqrt{n})$.

No attempt was made to optimize the constants. A few technical results will be needed in the sequel.

LEMMA 3.3 *For* $p \geq 0, n \geq 1$, *and arbitrary dimension* k,

$$\left(\frac{1}{1+p}\right)^{\lfloor p\rfloor+1} \frac{\Gamma(p+1)}{n^{p-1}} \leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} |R_i|^p\right\} \leq \frac{4\Gamma(p+1)}{n^{p-1}},$$

for all n.

PROOF. Let $V_1, \ldots, V_{n+1}$ be the spacings induced by n independent uniformly distributed random variables on $[0,1]$. It is known that the spacings $V_i \stackrel{\mathcal{L}}{=} \text{beta}(1,n)$. Thus, by lemma 3.1, with $\mathrm{B}(s,t) = \frac{\Gamma(s)\Gamma(t)}{\Gamma(s+t)}$,

$$\begin{aligned}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} |R_i|^p\right\} &= \mathbf{E}\left\{\sum_{i=1}^{n+1} V_i^p\right\} = \sum_{i=1}^{n+1}\int_0^1 v^p \frac{(1-v)^{n-1}}{\mathrm{B}(1,n)}dv \\ &= (n+1)\frac{\mathrm{B}(p+1,n)}{\mathrm{B}(1,n)} = \Gamma(p+1)\frac{\Gamma(n+2)}{\Gamma(p+n+1)}.\end{aligned}$$

Now, as $\Gamma(x+1) = x\Gamma(x)$ for any $x > 0$, and for any natural number n and any $s \in [0,1]$, $n^{1-s} \leq \Gamma(n+1)/\Gamma(n+s) \leq (n+1)^{1-s}$ (see Mitrinović, 1970), then

$$\begin{aligned}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} |R_i|^p\right\} &= \Gamma(p+1)(n+1)\frac{\Gamma(n+1)}{(n+p)\cdots(n+p-\lfloor p\rfloor)\Gamma(n+p-\lfloor p\rfloor)} \\ &\leq \frac{\Gamma(p+1)(n+1)^{2-p+\lfloor p\rfloor}}{n^{\lfloor p\rfloor+1}} \\ &= \frac{\Gamma(p+1)}{n^{p-1}}\left(\frac{n+1}{n}\right)^{2+\lfloor p\rfloor-p} \\ &\leq \frac{4\Gamma(p+1)}{n^{p-1}}.\end{aligned}$$

Now, for the lower bound, note that

$$\begin{aligned}
\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}(X_iY_i)^p\right\} &= \Gamma(p+1)(n+1)\frac{\Gamma(n+1)}{(n+p)\cdots(n+p-\lfloor p\rfloor)\Gamma(n+p-\lfloor p\rfloor)} \\
&\geq \frac{\Gamma(p+1)}{n^{p-1}}\frac{n^{\lfloor p\rfloor+1}}{(n+p)\cdots(n+p-\lfloor p\rfloor)} \\
&\geq \frac{\Gamma(p+1)}{n^{p-1}}\left(\frac{n}{n+p}\right)^{\lfloor p\rfloor+1} \\
&\geq \frac{\Gamma(p+1)}{n^{p-1}}\left(\frac{1}{1+p}\right)^{\lfloor p\rfloor+1}.
\end{aligned}$$

□

LEMMA 3.4 *In a random squarish 2-d tree, for every $q \geq 1$,*

$$\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} Y_i^q\right\} \leq \begin{cases} \frac{8}{1-q/2}n^{1-q/2}, & \text{for } q\in[1,2); \\ 8e\log n, & \text{for } 2-\frac{2}{\log n}\leq q\leq 2; \\ \frac{5\Gamma(q/2+1)}{q/2-1}\left(\frac{q}{2}-\frac{1}{n^{q/2-1}}\right), & \text{for } q>2, \end{cases}$$

and for $q\in[1,2)$,

$$\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} Y_i^q\right\} \geq \left(\frac{1}{q/2+1}\right)^{\lfloor q/2\rfloor+1}\Gamma(q/2+1)n^{1-q/2}.$$

The same result holds for $\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i^q\right\}$.

PROOF. Let $r>1$, and define $S_r^{(q)} = \sum_{i\in\mathcal{F}_r} Y_i^q$. Note that, given $U_1,\ldots,U_r$, $S_{r+1}^{(q)} - S_r^{(q)}$ is distributed as Y^q when $X>Y$ and as $Y^q(U^q+(1-U)^q-1)$ when $X\leq Y$, where U is a uniform $[0,1]$ random variable, and (X,Y) are the dimensions of the rectangle split when U_{r+1} is added. Thus,

$$\mathbf{E}\left\{S_{r+1}^{(q)} - S_r^{(q)}\right\} = \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} X_iY_i\left(1_{[X_i>Y_i]}Y_i^q + 1_{[X_i<Y_i]}Y_i^q\left(U^q+(1-U)^q-1\right)\right)\right\}.$$

Notice that $U^q+(1-U)^q-1\leq 0$ for $q\geq 1$, and as $\min\{a,b\}\leq\sqrt{ab}$, for $a,b\geq 0$, then by lemmas 3.1 and 3.3,

$$\begin{aligned}
\mathbf{E}\left\{S_{r+1}^{(q)} - S_r^{(q)}\right\} &\leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} X_iY_i\left(1_{[X_i>Y_i]}Y_i^q\right)\right\} \\
&\leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}(X_iY_i)^{q/2+1}\right\} \leq \frac{4\Gamma(q/2+2)}{r^{q/2}}.
\end{aligned}$$

By summing the differences we get,

$$\begin{aligned}
\mathbf{E}\left\{S_n^{(q)}\right\} &= \mathbf{E}\left\{\sum_{r=1}^{n-1}\left(S_{r+1}^{(q)} - S_r^{(q)}\right) + S_1^{(q)}\right\} \\
&\leq \sum_{r=1}^{n-1} \frac{4\Gamma(q/2+2)}{r^{q/2}} + 2 \\
&\leq 2 + 4\Gamma(q/2+2)\left(1 + \int_1^{n-1} \frac{1}{x^{q/2}} dx\right) \\
&\leq \begin{cases} 10 + \frac{4\Gamma(q/2+2)}{1-q/2}(n^{1-q/2}-1); & (q \in [1,2)) \\ 5\Gamma(q/2+2) + \frac{4\Gamma(q/2+2)}{q/2-1}(1-n^{1-q/2}) & (q>2) \end{cases} \\
&\leq \begin{cases} \frac{8}{1-q/2} n^{1-q/2}; & (q \in [1,2)) \\ \frac{5\Gamma(q/2+2)}{q/2-1}\left(\frac{q}{2} - n^{1-q/2}\right) & (q>2). \end{cases}
\end{aligned}$$

Because $\frac{8}{1-q/2} n^{1-q/2}$, as a function of q, reaches its minimum at $q_0 = 2(1 - 1/\log n)$, and $\mathbf{E}\left\{S_n^{(q)}\right\}$ is a decreasing function of q, we have that $\mathbf{E}\left\{S_n^{(q)}\right\} \leq 8e \log n$, for $q_0 \leq q \leq 2$. The result for $\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i^q\right\}$ can be obtained similarly just by replacing the y-lengths for the x-lengths in the appropriate places.

Now, for the lower bound, note that as the X_i's and the Y_i's are identically distributed:

$$\begin{aligned}
\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} Y_i^q\right\} &= \frac{1}{2}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}(Y_i^q + X_i^q)\right\} \\
&\geq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}(Y_i X_i)^{q/2}\right\} \\
&\geq \left(\frac{1}{q/2+1}\right)^{\lfloor q/2\rfloor+1} \frac{\Gamma(q/2+1)}{n^{q/2-1}},
\end{aligned}$$

by lemma 3.3, for $q \in [1,2)$. □

PROOF OF THEOREM 3.3. Note that the lower bound follows directly from lemma 3.4, as $\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} Y_i\right\}$ is less than $\mathbf{E}\left\{\sum_{i=1}^{2n+1} Y_i\right\}$. For the upper bound we use the same technique as in the proof of lemma 3.4. Let $S_n = \sum_{i=1}^{2n+1} Y_i$. Note that as the sum is over all the rectangles generated by $U_1, \ldots, U_n$, we

have now that for $r \geq 1$, as X_i and Y_i are identically distributed,

$$\begin{aligned}\mathbf{E}\left\{S_{r+1} - S_r\right\} &= \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} X_iY_i\left(1_{[X_i>Y_i]}2Y_i + 1_{[X_i<Y_i]}(Y_iU + Y_i(1-U))\right)\right\} \\ &= 3\,\mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} X_iY_i^2\right\}\end{aligned}$$

where $U \stackrel{\mathcal{L}}{=} \text{Uniform}[0,1]$, and independent of all $U_1,\ldots,U_n$. Let $q \in (1,2)$ and $p > 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$, then by Hölder's inequality used twice,

$$\begin{aligned}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} X_iY_i^2\right\} &\leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}(X_iY_i)^p\right\}^{1/p}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_r} Y_i^q\right\}^{1/q} \\ &\leq \left(\frac{4\Gamma(p+1)}{r^{p-1}}\right)^{1/p}\left(\frac{8}{1-q/2}\frac{1}{r^{q/2-1}}\right)^{1/q}\end{aligned}$$

by lemmas 3.3 and 3.4. Take $p = 3$, $q = 3/2$. and verify that the upper bound is not more than $24^{1/3}32^{2/3}/\sqrt{r} < 30/\sqrt{r}$. By summing the differences we finally obtain

$$\mathbf{E}\left\{\sum_{i=1}^{2n+1} Y_i\right\} \leq \frac{5}{2} + 90\sum_{r=1}^{n-1}\frac{1}{\sqrt{r}} \leq \frac{5}{2} + 90(2\sqrt{n-1}\ -1) \leq 180\sqrt{n}.$$

For the lower bound, set $q = 1$ in the lower bound of lemma 3.4. The result for $\mathbf{E}\left\{\sum_{i=1}^{2n+1} X_i\right\}$ can be obtained similarly just by replacing the y-lengths for the x-lengths in the appropriate places. □

3.4 The k-dimensional case.

In this section, we obtain the k dimensional generalization of the results in the previous section by induction. Given $U_1,\ldots,U_n$, we define for each $R_i \in \mathcal{R}_n$, $X_i^* = \max_{j=1,\ldots,k} X_{ij}$ and j_i^* as the index $j \in \{1,\ldots,n\}$ for which $X_{ij} = X_i^*$. Note that j_i^* is unique with probability 1. Our main result generalizes theorem 3.3 and establishes the expected time optimality of random squarish k-d trees.

THEOREM 3.4. *Consider a random squarish k-d tree. For $\ell \in \{1,\ldots,k-1\}$, there exist $C, C' > 0$ such that*

$$C' n^{1-\frac{\ell}{k}} \leq \mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j\in I} X_{ij}\right\} \leq C n^{1-\frac{\ell}{k}},$$

for any $I \subseteq \{1,\ldots,k\}$ of cardinality ℓ and all $n \in \mathbb{N}$. In particular, by proposition 2.1 the expected time of a random partial match query with s specified coordinates is $\Theta(n^{1-s/k})$.

We prove the following lemma that allows us to prove the lower bound in the previous theorem.

LEMMA 3.5 *Let $\ell \in \{1,\ldots,k\}$, then for every $x_1,\ldots,x_k > 0$,*

$$\left(\prod_{j=1}^{k} x_j\right)^{\frac{1}{k}} \leq \max_{I:\ I\subseteq\{1,\ldots,k\}\atop |I|=\ell} \left(\prod_{j\in I} x_j\right)^{\frac{1}{\ell}}.$$

PROOF. Let I^* be the subset of $\{1,\ldots,k\}$ of cardinality ℓ for which the maximum above is reached. It suffices to observe that,

$$\left(\prod_{j=1}^{k} x_j\right)^{\ell} = \prod_{s=1}^{k}\left(\prod_{j=s}^{s+\ell-1} x_j\right) \leq \prod_{s=0}^{k-1}\left(\prod_{j\in I^*} x_j\right) = \left(\prod_{j\in I^*} x_j\right)^{k},$$

where the subindice j must be understood as (j mod k), if $j > k$. □

PROPOSITION 3.1 *Let $I \subseteq \{1,\ldots,k\}$ of cardinality $\ell \in \{1,\ldots,k\}$ and $p \in [1, \frac{k}{\ell})$, then there are positive constants C and C' such that*

$$C' n^{1-p\frac{\ell}{k}} \leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}\left(\prod_{j\in I} X_{ij}\right)^{p}\right\} \leq C n^{1-p\frac{\ell}{k}},$$

for all $n \in \mathbb{N}$.

PROOF. For $I \subseteq \{1,\ldots,k\}$ with $|I| = \ell$, we define

$$S_r^{I,p} = \sum_{i\in\mathcal{F}_r}\left(\prod_{j\in I} X_{ij}\right)^{p}.$$

We first look at the upper bound. We define recursively the constants $C_k(\ell, p)$ for any integer $k > 0$, $\ell \in \{1, \ldots, k\}$ and real number $p \in [1, \frac{k}{\ell})$ as follows,

$$C_k(\ell, p) = \begin{cases} 4\Gamma(p+1), & \text{if } \ell = k; \\ (k-\ell)\left(\frac{1}{1-\frac{p\ell}{k}}\right) C_k(k, \tilde{q})^{1/\tilde{q}} C_k(\ell+1, p\tilde{p}\ell/(\ell+1))^{1/\tilde{p}} + 2 & \text{if } \ell < k, \end{cases}$$

where $\tilde{p}, \tilde{q} > 1$ depend on p, k and ℓ, they are such that $\frac{1}{\tilde{p}} + \frac{1}{\tilde{q}} = 1$, and $1 \leq p\tilde{p}\frac{\ell}{\ell+1} < \frac{k}{\ell+1}$. For the sake of clarity we will choose $\tilde{p}$ later.

For $\ell \in \{2, \ldots, k\}$, we define the hypothesis $\mathcal{H}_\ell$ stating that the upper bound holds for all $n \in \mathbb{N}$, all $I \subseteq \{1, \ldots, k\}$ such that $|I| = \ell$, and all $p \in [1, \frac{k}{\ell})$, with constant $C_k(\ell, p)$. We prove $\mathcal{H}_\ell$ with an inductive argument. First, note that $\mathcal{H}_k$ holds by lemma 3.3. Assuming that $\mathcal{H}_\ell$ is true, we prove $\mathcal{H}_{\ell-1}$. Let $I \subseteq \{1, \ldots, k\}$ such that $\ell - 1 = |I| \geq 1$, and $p \in [1, \frac{k}{\ell-1})$. Then for any integer $r \geq 1$ we have,

$$\mathbf{E}\left\{S_{r+1}^{I,p} - S_r^{I,p} | U_1, \ldots, U_r\right\} = \sum_{i \in \mathcal{F}_r} \left(\prod_{j=1}^{k} X_{ij}\right) \left\{ 1_{[j_i^* \notin I]} \left(\prod_{j \in I} X_{ij}\right)^p + 1_{[j_i^* \in I]} \left(\prod_{j \in I} X_{ij}\right)^p \int_0^1 (x^p + (1-x)^p - 1) dx \right\},$$

as we are using the longest edge cut method. Since $\int_0^1 (x^p + (1-x)^p - 1) dx \leq 0$ for any $p \geq 1$, we can drop the second term above and take expected values so that,

$$\mathbf{E}\left\{S_{r+1}^{I,p} - S_r^{I,p}\right\} \leq \sum_{t \notin I} \mathbf{E}\left\{ \sum_{i \in \mathcal{F}_r} \left(\prod_{j=1}^{k} X_{ij}\right) 1_{[j_i^* = t]} \left(\prod_{j \in I} X_{ij}\right)^p \right\}.$$

Let us denote by $E(t)$ the expected value of the t^{th} term above. Observe that $1_{[j_i^* = t]} X_{ij} \leq X_{ij}^{\frac{\ell-1}{\ell}} X_{it}^{\frac{1}{\ell}}$. Thus we can bound each $E(t)$ as follows,

$$E(t) \leq \mathbf{E}\left\{ \sum_{i \in \mathcal{F}_r} \left(\prod_{j=1}^{k} X_{ij}\right) \left(\prod_{j \in I \cup \{t\}} X_{ij}\right)^{\frac{\ell-1}{\ell} p} \right\}.$$

Now, for any $\tilde{p}, \tilde{q} > 1$ such that $\frac{1}{\tilde{p}} + \frac{1}{\tilde{q}} = 1$, we have by applying Hölder's inequality twice that,

$$E(t) \leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}\left(\prod_{j=1}^{k} X_{ij}\right)^{\tilde{q}}\right\}^{\frac{1}{\tilde{q}}} \mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}\left(\prod_{j\in I\cup\{t\}} X_{ij}\right)^{\frac{\ell-1}{\ell}p\tilde{p}}\right\}^{\frac{1}{\tilde{p}}}.$$

We can apply hypothesis $\mathcal{H}_\ell$ to bound the second term above, if we can choose $\tilde{p} > 1$ such that $p\tilde{p}\frac{\ell-1}{\ell} \in [1, k/\ell)$. Note that $\frac{k}{p(\ell-1)} > 1$, as $p \in [1, \frac{k}{\ell-1})$. Let us define $\tilde{p} = \max\left\{\sqrt{k/p(\ell-1)}, \frac{\ell}{(\ell-1)p}\right\}$, so that $\tilde{p} > 1$, yet $1 \leq p\tilde{p}\frac{\ell-1}{\ell} < \frac{k}{\ell}$. This completely defines the constant $C_k(\ell, p)$. We can therefore use hypothesis $\mathcal{H}_\ell$ and obtain,

$$\begin{aligned} E(t) &\leq \left(\frac{C_k(k,\tilde{q})}{r^{\tilde{q}-1}}\right)^{1/\tilde{q}} \left(\frac{C_k(\ell, p\tilde{p}(\ell-1)/\ell)}{r^{\frac{\ell-1}{k}p\tilde{p}-1}}\right)^{1/\tilde{p}} \\ &= \frac{C_k(k,\tilde{q})^{1/\tilde{q}} C_k(\ell, p\tilde{p}(\ell-1)/\ell)^{\frac{1}{\tilde{p}}}}{r^{\frac{\ell-1}{k}p}}. \end{aligned}$$

We can thus bound the differences as follows,

$$\mathbf{E}\left\{S_{r+1}^{I,p} - S_r^{I,p}\right\} \leq \sum_{t\notin I} E(t) \leq \frac{(k-\ell+1)C_k(k,\tilde{q})^{1/\tilde{q}} C_k(\ell, p\tilde{p}(\ell-1)/\ell)^{1/\tilde{p}}}{r^{\frac{\ell-1}{k}p}}.$$

Since $p < \frac{k}{\ell-1}$, we have that $\sum_{r=1}^{n} \frac{1}{r^{p\frac{\ell-1}{k}}} \leq \frac{1}{1-p\frac{\ell-1}{k}} \frac{n}{n^{p\frac{\ell-1}{k}}}$. So, by summing the differences we get,

$$\mathbf{E}\left\{S_n^{I,p}\right\} \leq [C_k(\ell-1,p) - 2]\, n^{1-p\frac{\ell-1}{k}} + 2 \leq C_k(\ell-1,p) n^{1-\frac{p(\ell-1)}{k}}$$

as $\mathbf{E}\left\{S_1^{I,p}\right\} \leq 2$, for every $p \geq 1$, and any nonempty $I \subseteq \{1,\ldots,k\}$. Thus, hypothesis $\mathcal{H}_{\ell-1}$ is proved.

We now prove the lower bound. As we flip a perfect coin at the beginning of the process to choose the side of R_0 that we cut, all the coordinates $X_{i1},\ldots,X_{ik}$ of a rectangle R_i are exchangeable. So, denoting by $\mathcal{S}$ the set of all $I' \subseteq \{1,\ldots,k\}$ of cardinality ℓ, all the random variables $\sum_{i\in\mathcal{F}_n}\prod_{j\in I'} X_{ij}^p$ are equally distributed so that:

$$\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}\prod_{j\in I} X_{ij}^p\right\} = \frac{1}{|\mathcal{S}|}\mathbf{E}\left\{\sum_{I'\in\mathcal{S}}\sum_{i\in\mathcal{F}_n}\prod_{j\in I'} X_{ij}^p\right\}.$$

Then, by lemmas 3.3 and 3.2,

$$\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}\left(\prod_{j\in I}X_{ij}\right)^p\right\}\geq\frac{1}{|\mathcal{S}|}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}\left(\prod_{j=1}^{k}X_{ij}\right)^{\frac{p\ell}{k}}\right\}\geq C'\frac{n}{n^{\frac{p\ell}{k}}}.$$

□

We must note that by lemma 3.3, if $\ell=k$, then for any $p\geq 0$, there are positive constants C and C', depending on p such that the previous result holds. We are now ready to prove theorem 3.4.

PROOF OF THEOREM 3.4. The lower bound follows immediately from the previous proposition. For any subset $I\subseteq\{1,\ldots,k\}$ of cardinality $\ell\in\{1,\ldots,k-1\}$, we define:

$$S_n^I=\sum_{i=1}^{2n}\prod_{j\in I}X_{ij}.$$

As we are using the longest edge cut method we have that,

$$\begin{aligned}\mathbf{E}\left\{S_{r+1}^I-S_r^I|U_1,\ldots,U_n\right\}&=\sum_{i\in\mathcal{F}_r}\prod_{j=1}^{k}X_{ij}\left\{1_{[j_i^*\notin I]}2\prod_{j\in I}X_{ij}+1_{[j_i^*\in I]}\prod_{j\in I}X_{ij}\right\}\\&\leq 3\sum_{i\in\mathcal{F}_r}\prod_{j=1}^{k}X_{ij}\prod_{j\in I}X_{ij}.\end{aligned}$$

We choose now $p=\sqrt{k/\ell}$, $q=1/(1-\sqrt{\ell/k})$, so that $\frac{1}{p}+\frac{1}{q}=1$, and apply Hölder inequality with these values to get,

$$\mathbf{E}\left\{S_{r+1}^I-S_r^I\right\}\leq 3\,\mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}\left(\prod_{j=1}^{k}X_{ij}\right)^p\right\}^{1/p}\mathbf{E}\left\{\sum_{i\in\mathcal{F}_r}\left(\prod_{j\in I}X_{ij}\right)^q\right\}^{1/q}.$$

Then by lemma 3.3 and proposition 3.2, there exists a positive constant C depending upon ℓ and k such that

$$\mathbf{E}\left\{S_{r+1}^I-S_r^I\right\}\leq\frac{C}{r^{\frac{\ell}{k}}}.$$

We add the differences to get

$$\mathbf{E}\left\{S_n^I\right\} \leq C\left(\sum_{r=1}^{n} \frac{1}{r^{\frac{\ell}{k}}}\right) + 2 \leq \frac{C}{1-\frac{\ell}{k}}\left(\frac{n}{n^{\frac{\ell}{k}}}\right) + 2.$$

□

3.5 Conclusions

Note that off-line one may construct a median k-d tree by splitting each time about the median, thus obtaining a perfectly balanced binary tree, in which search takes $\Theta(\log n)$ worst-case time, and a partial match query takes worst-case time $O(n^{1-1/k} + N)$, where N is the number of points returned (see for example, Lee and Wong, 1977). Therefore we see that k-d trees are not optimal even in an average sense for solving range search. The elongated rectangles in the partition generated using k-d trees explain its poor performance. For squarish k-d trees, however, we have shown that they behave optimally in an expected sense. For instance, for 2-d trees we have that the expected time complexity of partial match, when specifying one attribute, is $\Theta\left(n^{\frac{\sqrt{17}-3}{2}}\right) = \Theta\left(n^{0.561552\ldots}\right)$, whereas for 2-d squarish trees it is $\Theta(\sqrt{n})$. Relaxed k-d trees have even worse expected complexity time, as for example in dimension 2 they have expected complexity $\Theta\left(n^{\frac{1+\sqrt{5}}{2}}\right) = \Theta\left(n^{0.618034\ldots}\right)$, though they have the advantage of supporting deletions easily.

Chapter 4
Orthogonal Range Search

In this chapter, we obtain tight upper bounds for the expected time complexity for Bentley's orthogonal range search algorithm. We present theorems showing that random squarish k-d trees are superior to random k-d trees for any kind of random orthogonal range search where the dimensions of the query region are allowed to depend upon n, the number of data points, in an arbitrary manner. We also generalize these results for arbitrary convex range search problems in dimension 2. The novelty here is that the dimensions of the search regions are allowed to depend upon n in an arbitrary manner.

4.1 Orthogonal range search and k-d trees

Let us first state the theorem about the behavior of range search when using k-d trees.

THEOREM 4.1. *Given is a random k-d tree based on i.i.d. random variables $U_1, \ldots, U_n$, distributed uniformly on $[0,1]^k$. Let Q be a random query rectangle of dimensions $\Delta_1 \times \cdots \times \Delta_k$ (which are deterministic functions of n taking values in $[0,1]$), with center at Z which is uniformly distributed on $[0,1]^k$, and independent of $U_1, \ldots, U_n$. Let N_n be the number of comparisons that Bentley's orthogonal range search algorithm performs. Then, there exist constants $\gamma', \gamma > 0$ depending upon k only such that*

$$\gamma' \leq \frac{\mathbf{E}\{N_n\}}{\left(\log n + \sum_{\substack{S \subseteq \{1,\ldots,k\} \\ |S|<k}} \left(\prod_{j \notin S} \Delta_j\right) n^{\alpha(|S|/k)}\right)} \leq \gamma.$$

To prove the previous theorem we will need the following result.

PROPOSITION 4.1 *Given is a random k-d tree based on i.i.d. random variables $U_1, \ldots, U_n$, distributed uniformly on $[0,1]^k$. Let X_{ij} be the length of the j^{th} side of rectangle $R_i \in \mathcal{R}_n$. Then, there is a constant $C > 0$, depending on k only, such that*

$$\mathbf{E}\left\{\sum_{i=1}^{2n+1} 1_{\left[\max_{j\in\{1,\ldots,k\}} X_{ij} \geq \frac{1}{2}\right]}\right\} \leq C.$$

PROOF. For $\ell \geq 1$, let $X^{(\ell)}$ be the product of $\lfloor \ell/k \rfloor$ independent Uniform$[0,1]$ random variables. Then,

$$\begin{aligned}
\mathbf{E}\left\{\sum_{i=1}^{2n+1} 1_{\left[\max_{j\in\{1,\ldots,k\}} X_{ij} \geq \frac{1}{2}\right]}\right\} &\leq \mathbf{E}\left\{\sum_{i=1}^{2n+1}\sum_{j=1}^{k} 1_{\left[X_{ij} \geq \frac{1}{2}\right]}\right\} \\
&\leq \sum_{\ell=1}^{\infty} 2^\ell k\, \mathbf{E}\left\{1_{\left[X^{(\ell)} \geq \frac{1}{2}\right]}\right\} \\
&\leq \sum_{\ell=1}^{\infty} 2^\ell k\, \mathbf{E}\left\{\left(X^{(\ell)}\right)^p\right\} 2^p \\
&\leq k2^p(p+1)\sum_{\ell=1}^{\infty}\left(\frac{2}{(p+1)^{\frac{1}{k}}}\right)^\ell,
\end{aligned}$$

for any $p \geq 0$. The last expression is finite, for example, if we take $p = 2^k$, as $k \geq 2$. □

PROOF OF THEOREM 4.1. Note that given $U_1, \ldots, U_n$, the probability that Q intersects R_i is the probability that Z has some coordinate Z_j that is within distance $\Delta_j/2$ of R_i, and this probability is clearly bounded by the volume of R_i expanded by Δ_j in the j-th direction, for all j. Thus,

$$\begin{aligned}
\mathbf{E}\{N_n\} &\leq \mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j=1}^{k}(X_{ij}+\Delta_j)\right\} \\
&= \sum_{S\subseteq\{1,\ldots,k\}}\left(\prod_{j\notin S}\Delta_j\right)\mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j\in S}X_{ij}\right\} \\
&\leq C\sum_{S\subseteq\{1,\ldots,k\}:|S|<k}\left(\prod_{j\notin S}\Delta_j\right)n^{\alpha(|S|/k)} + 2H_{n+1} - 1
\end{aligned}$$

for some constant $C > 0$ and for all n large enough, by theorem 3.1 and lemma 3.2. For the lower bound notice that

$$\begin{aligned}
\mathbf{E}\{N_n\} &\geq \mathbf{E}\left\{\sum_{i=1}^{2n+1} \mathbf{1}_{[Q\cap R_i\neq\emptyset]}\mathbf{1}_{[\forall j\in\{1,\ldots,k\}:X_{ij}<1/2]}\right\} \\
&\geq \mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j=1}^{k}\left(X_{ij}+\frac{\Delta_j}{2}\right)\mathbf{1}_{[\forall j\in\{1,\ldots,k\}:X_{ij}<1/2]}\right\} \\
&= \mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j=1}^{k}\left(X_{ij}+\frac{\Delta_j}{2}\right)\right\} \\
&\quad - \mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j=1}^{k}\left(X_{ij}+\frac{\Delta_j}{2}\right)\mathbf{1}_{[\exists j\in\{1,\ldots,k\}:X_{ij}\geq 1/2]}\right\} \\
&= \sum_{S\subseteq\{1,\ldots,k\}}\prod_{j\notin S}\frac{\Delta_j}{2}\,\mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j\in S}X_{ij}\right\} \\
&\quad - \sum_{S\subseteq\{1,\ldots,k\}}\prod_{j\notin S}\frac{\Delta_j}{2}\,\mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j\in S}X_{ij}\mathbf{1}_{[\exists j\in\{1,\ldots,k\}:X_{ij}>1/2]}\right\}.
\end{aligned}$$

We can bound the second term above for any given $S \subseteq \{1,\ldots,k\}$ as follows:

$$\begin{aligned}
\mathbf{E}\left\{\sum_{i=1}^{2n+1}\prod_{j\in S}X_{ij}\mathbf{1}_{[\exists j\in\{1,\ldots,k\}:X_{ij}>1/2]}\right\} &\leq \mathbf{E}\left\{\sum_{i=1}^{2n+1}\mathbf{1}_{\left[\max_{j\in\{1,\ldots,k\}}X_{ij}\geq\frac{1}{2}\right]}\right\} \\
&\leq C',
\end{aligned}$$

by the previous proposition. The result follows again by theorem 3.1 and lemma 3.2. □

We must note that the arguments to prove the lower bound indeed apply to any range search algorithm for solving the range search problem using k-d trees.

TWO-DIMENSIONAL SPECIAL CASE. For $k = 2$, as $\alpha(1/2) = \frac{\sqrt{17}-3}{2} \approx 0.5616$, we see that the expected complexity bound is

$$O\left(\log n + n^{\frac{\sqrt{17}-3}{2}}(\Delta_1+\Delta_2) + n\Delta_1\Delta_2\right).$$

The first term accounts for complexity due to search in a tree of height $\log n$. The last term is a volume term, approximately equal to the number of points

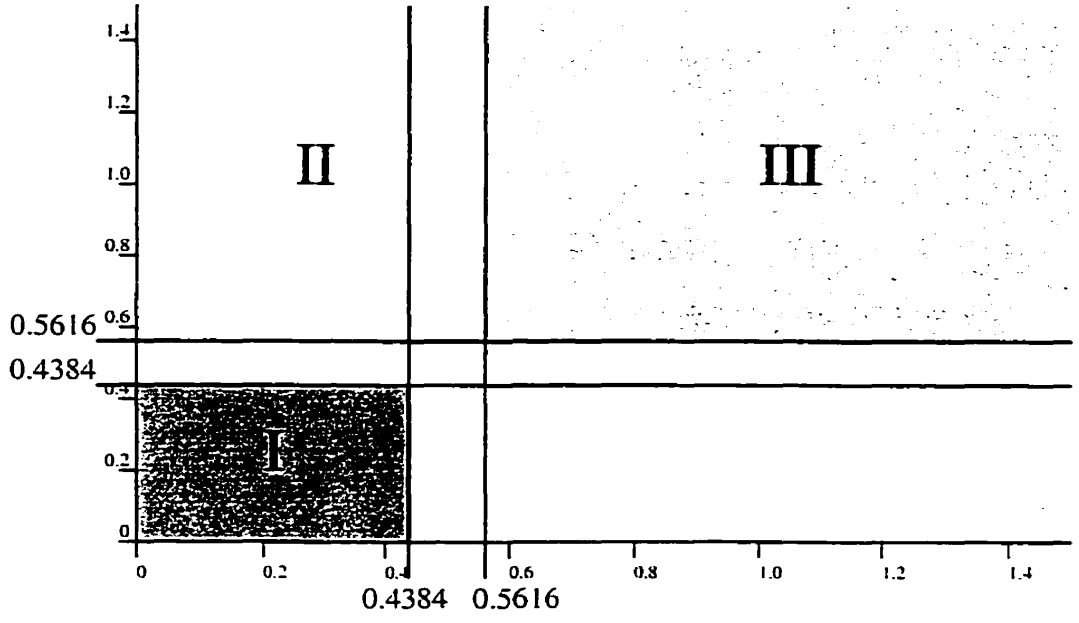

FIGURE 4.1. The complexity regions: in I, the output size dominates. In II, the 1-d complexity term is largest, and III is like point search.

returned by the query. Both are unavoidable. The middle term is due to complexity related to the perimeter of the query rectangle as a long perimeter cuts many rectangles in the partition. In case $\Delta_1 = 1/n^a$ and $\Delta_2 = 1/n^b$ with $a, b \geq 0$, figure 4.1 below shows the regions of the (a, b) plane in which each of the terms dominates. The perimeter term dominates in the white region, the volume term dominates in the dark region, and the search term $(\log n)$ dominates in the light region. Point search corresponds to $a = b = \infty$, and a partial match query corresponds to $a = 0, b = \infty$ or vice versa, which falls plainly in the white region. Put differently, we have

$$\mathbf{E}\{N_n\} = \begin{cases} O(\log n) & \text{if } \min(a,b) \geq \alpha(1/2) = \frac{\sqrt{17}-3}{2} \\ O(n^{1-a-b}) & \text{if } \max(a,b) \leq 1-\alpha(1/2) = \frac{5-\sqrt{17}}{2} \\ O(n^{\frac{\sqrt{17}-3}{2}-\min(a,b)}) & \text{otherwise.} \end{cases}$$

4.2 Orthogonal range search and squarish k-d trees

In this section, we obtain tight upper bounds for the expected complexity for Bentley's range search algorithm when using as underlying data structure

squarish k-d trees. Theorem 4.2 below then states that random squarish k-d trees are superior to random k-d trees for any kind of random orthogonal range search.

THEOREM 4.2. *Let Q be a random query rectangle of dimensions $\Delta_1 \times \cdots \times \Delta_k$ (which are deterministic functions of n taking values in $[0,1]$), with center at Z which is uniformly distributed on $[0,1]^k$, and independent of the k-d tree. Let N_n be the number of comparisons that Bentley's orthogonal range search algorithm performs. Then, there exist constants $\gamma > \gamma' > 0$ depending upon k only such that*

$$\gamma' \leq \frac{\mathbf{E}\{N_n\}}{\left(\log n + \sum_{\substack{S\subseteq\{1,\ldots,k\}\\ 0\leq|S|<k}} \prod_{j\notin S} \Delta_j n^{1-\frac{|S|}{k}}\right)} \leq \gamma .$$

We can rewrite the previous result as follows,

$$\mathbf{E}\{N_n\} \leq \gamma \left(n\prod_{i=1}^{k}\Delta_j + \sum_{\ell=1}^{k-1} n^{1-\frac{\ell}{k}} \sum_{\substack{S\subseteq\{1,\ldots,k\}\\ |S|=\ell}} \prod_{j\notin S}\Delta_j + \log n \right),$$

and therefore by allowing any r of the Δ_j's to be zero, the term that dominates the previous bound is,

$$n^{1-\frac{r}{k}} \sum_{S;|S|=r} \prod_{j\notin S} \Delta_j .$$

For example, when $k=2$, $\Delta = \Theta(1/n^a)$, and $\Delta' = \Theta(1/n^b)$, then

$$\mathbf{E}\{N_n\} \leq \gamma \left(n^{1-a-b} + n^{\frac{1}{2}-a} + n^{\frac{1}{2}-b} + \log n \right).$$

By looking at the different regions in the a-b plane we obtain,

$$\mathbf{E}\{N_n\} \leq \begin{cases} \Theta(\log n), & \text{for } a \geq 1/2 \text{ and } b \geq 1/2; \\ \Theta(\max\{n^{1/2-a} n^{1/2-b}\}), & \text{for } a > 1/2, b < 1/2, \text{ or } a < 1/2, b > 1/2; \\ \Theta(n^{1-a-b}), & \text{for } a \leq 1/2, b \leq 1/2. \end{cases}$$

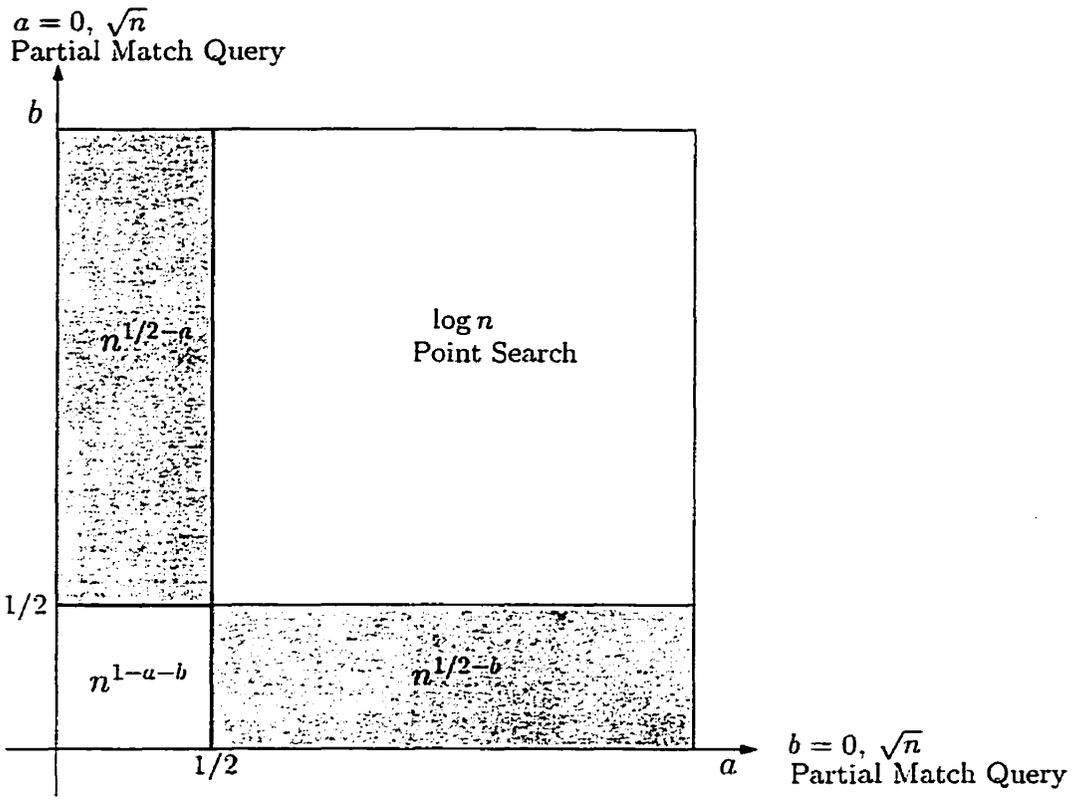

FIGURE 4.2. The complexity regions for $\Delta = \Theta(1/n^a)$ and $\Delta' = \Theta(1/n^b)$.

Note that if $a = 0$ and $b \geq 1/2$, or $b = 0$ and $a \geq 1/2$, we recover the expected complexity time of the random partial match query problem.

PROPOSITION 4.2 *Let $U_1, \ldots, U_n$ be independent and uniformly distributed over $[0,1]^k$ random variables, let X_i^* be the longest side of the i^{th} rectangle generated by $U_1, \ldots, U_n$. Then, for all $n \geq 0$,*

$$\mathbf{E}\left\{\sum_{i \in \mathcal{F}_n} 1_{[X_i^* > \frac{1}{2}]}\right\} \leq 2^{4k-3}.$$

PROOF. Note that $\mathbf{E}\left\{\sum_{i \in \mathcal{F}_n} 1_{[X_i^* > \frac{1}{2}]}\right\} \leq 2^k \mathbf{E}\left\{\sum_{i \in \mathcal{F}_n} \prod_{j \in I_i} X_{ij}\right\}$, where $I_i = \{j : X_{ij} > \frac{1}{2}\}$. Define $S_n = \sum_{i \in \mathcal{F}_n} \left(\prod_{j: X_{ij} > \frac{1}{2}} 8 X_{ij}\right)$. We are going to prove that $\mathbf{E}\{S_n\}$ is decreasing so that for $n \geq 1$,

$$\mathbf{E}\left\{\sum_{i \in \mathcal{F}_n} 1_{[X_i^* > \frac{1}{2}]}\right\} \leq 2^{k-3} \mathbf{E}\{S_n\} \leq 2^{k-3} \mathbf{E}\{S_0\} = 2^{4k-3}.$$

To show $\mathbf{E}\{S_n\} \leq \mathbf{E}\{S_0\}$, we look at the differences once again. Set $P_i = \prod_{j \in I_i} 8X_{ij}$. Then,

$$\begin{aligned} S_{r+1} - S_r = \sum_{i \in \mathcal{F}_n} |R_i| 1_{[X_i^* > \frac{1}{2}]} \Bigg\{ & -P_i + 1_{[XX_i^* > \frac{1}{2}]} \left(P_i X + 1_{[|I_i|>1]} \frac{P_i}{8X_i^*} \right) \\ & + 1_{[(1-X)X_i^* > \frac{1}{2}]} \left(P_i(1-X) + 1_{[|I_i|>1]} \frac{P_i}{8X_i^*} \right) \\ & + 1_{[XX_i^* \leq \frac{1}{2};\ (1-X)X_i^* \leq \frac{1}{2}]} \left(2 1_{[|I_i|>1]} \frac{P_i}{8X_i^*} \right) \Bigg\}, \end{aligned}$$

where $X \stackrel{\mathcal{L}}{=} \text{Uniform}[0,1]$, and it is independent of $U_1, \ldots, U_r$. Therefore,

$$\begin{aligned} \mathbf{E}\{S_{r+1} - S_r | U_1, \ldots, U_r\} & \leq \sum_{i \in \mathcal{F}_r} |R_i| 1_{[X_i^* > \frac{1}{2}]} P_i \left(-1 + \int_{\frac{1}{2X_i^*}}^1 (x + \frac{1}{4}) dx \right. \\ & \qquad \left. + \int_0^{1 - \frac{1}{2X_i^*}} ((1-x) + 1/4) dx + \int_{1 - \frac{1}{2X_i^*}}^{\frac{1}{2X_i^*}} 1/2 dx \right) \\ & = \sum_{i \in \mathcal{F}_r} |R_i| 1_{[X_i^* > \frac{1}{2}]} P_i \left(\frac{1}{4X_i^*} - \frac{1}{(2X_i^*)^2} \right) \\ & \leq 0. \end{aligned}$$

□

PROOF OF THEOREM 4.2. The proof follows exactly the same arguments as the proof of theorem 4.1, except that so as to prove the lower bound we use proposition 4.2 instead of proposition 4.1. □

4.3 Searching with convex sets

To perform a range search with a convex set C, we may also recursively descend the k-d tree, and visit all subtrees whose root rectangle has a nonempty intersection with C. In this section, to fix the ideas, we consider $k = 2$ only, although the generalizations to higher dimensions are straightforward. For a fixed convex set C, we let $\mathcal{E}_C$ denote the minimal ellipse containing C. Let the center of $\mathcal{E}_C$ be the origin. Let $\mathcal{E}_C$ have principal axes u and v, with u perpendicular to v. Let R_C be the smallest rectangle aligned with the axes

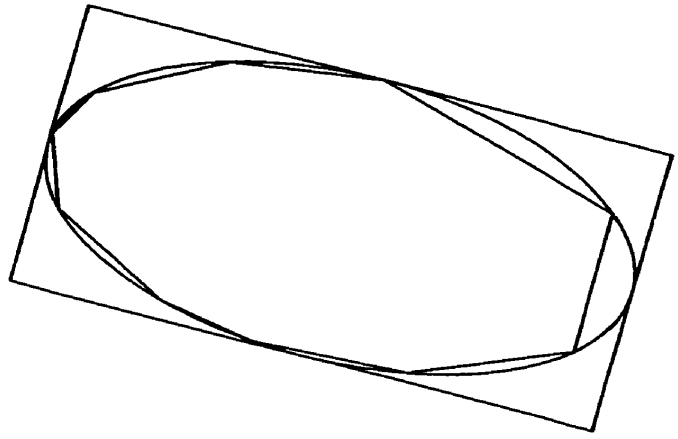

FIGURE 4.3. Construction used for convex sets.

(u, v) pair that contains $\mathcal{E}_C$ (and thus touches the ellipse in just four points). Let the dimensions of the rectangle R_C (and thus of $\mathcal{E}_C$) in the u and v directions be $\Delta > 0$ and $\Delta' > 0$ respectively. These dimensions are deterministic but may depend on n. A random range search is defined as a range search with convex set $Z + C$, the translation by Z (a uniformly distributed random variable on $[0,1]^k$) of C.

First we generalize theorems 4.1 and 4.2 to rotated rectangles. Let Q be a rectangle of size $\Delta \times \Delta'$ parallel to $[0,1]^2$ and centered at the origin. For $\phi \in [0, 2\pi)$, we define Q_ϕ as the rectangle resulting from rotating Q by ϕ about the origin.

THEOREM 4.3. *Let $U_1, \ldots, U_n$ be independent and uniform random variables over $[0,1]^2$, used to construct either a 2-d tree or a squarish 2-d tree, and let $\mathcal{R}_n$ be the corresponding partition into rectangles. Let Z be uniformly distributed over $[0,1]^2$, independent of the U_i's, and let N_n be the number of rectangles in $\mathcal{R}_n$ that intersect $Z + Q_\phi$ (and thus the complexity of range search with this set). If Q has dimensions $\Delta \times \Delta'$, then there is a universal constant $\gamma > 0$ (not depending upon n, Δ, Δ' or ϕ), such that*

$$\mathbf{E}\{N_n\} \leq \gamma\left(n\Delta\Delta' + (\Delta + \Delta')n^\alpha + \log n\right),$$

where $\alpha = \frac{\sqrt{17}-3}{2}$, if we used 2-d trees and $\alpha = 1/2$ if we use squarish 2-d trees.

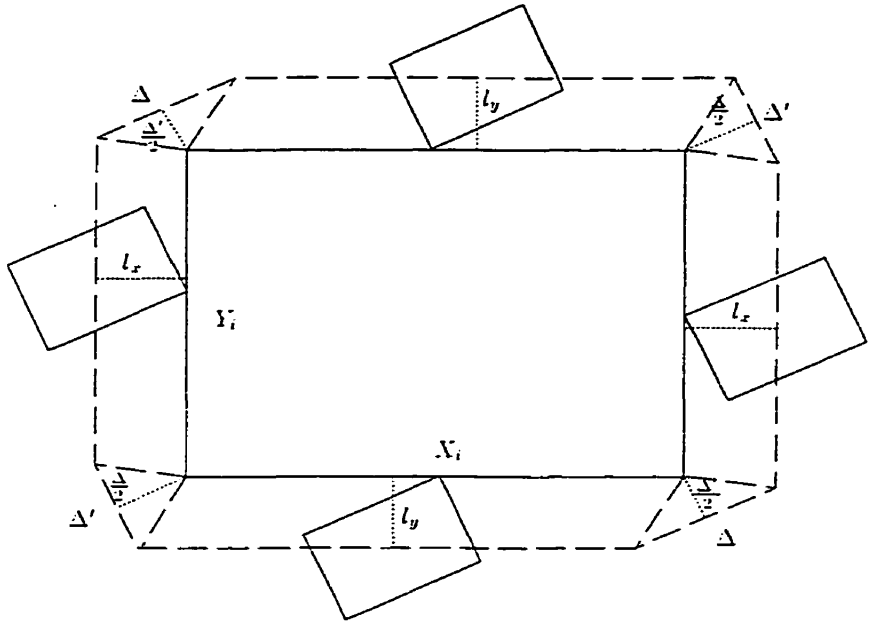

FIGURE 4.4. Areas in theorem 4.3.

PROOF. If a rectangle R_i in $\mathcal{R}_n$ has dimensions $X_i \times Y_i$, then $Z + Q_\phi$ intersects it if and only if Z falls in the octagon outlined in figure 4.4, where the tilted rectangles are various positions of the tilted query rectangle. It is easy to see that this octagon in turn is contained in the rectangle R_i extended on top and bottom by l_y (again, see figure 4.4) and on left and right by l_x. Using the same reasoning as in theorems 4.1 and 4.2, we note that given $U_1, \ldots, U_n$, the probability that $Z + Q_\phi$ intersects R_i, is bounded by $X_i Y_i + 2\max(l_x, l_y)(X_i + Y_i) + 2\Delta\Delta'$. Clearly, $\max(l_x, l_y) \leq (\Delta + \Delta')/\sqrt{2}$. Thus,

$$\begin{aligned}\mathbf{E}\{N_n\} &\leq \mathbf{E}\left\{\sum_{i=1}^{2n+1}(X_i Y_i)\right\} + \sqrt{2}(\Delta + \Delta')\,\mathbf{E}\left\{\sum_{i=1}^{2n+1}(X_i + Y_i)\right\} + (4n+2)\Delta\Delta' \\ &\leq 2H_{n+1} - 1 + c\sqrt{2}(\Delta + \Delta')n^\alpha + (4n+2)\Delta\Delta'\end{aligned}$$

for some constant $c > 0$ by lemma 3.2 and theorem 4.1 and 4.2, depending on whether we use 2-d trees or squarish 2-d trees.. Note in particular that the constant c does not depend upon ϕ. □

To prepare for the main result of this section, we use a fact from classical

geometry, stated here in its high-dimensional form. We use the following result by John (1948).

LEMMA 4.1 *Let S be any bounded set in $\mathbb{R}^k$ not contained in any linear subspace of it. Let $\mathcal{E}_S$ be the smallest ellipsoid containing S (called John's ellipsoid) and $\mathcal{E}'_S$ be the concentric and homothetic ellipsoid at the ratio of $\frac{1}{k}$. Then $\mathcal{E}'_S \subseteq \text{CH}(S) \subseteq \mathcal{E}_S$, where $\text{CH}(S)$ denotes the convex hull of S.*

In particular, John's result implies that $|\mathcal{E}_S| \leq k^k |\mathcal{E}'_S| \leq k^k |\text{CH}(S)|$. Let $\mathcal{E}$ be an ellipsoid with principal axes of lengths $a_1, \ldots, a_k$, and let B be the unit ball of $\mathbb{R}^k$. Then

$$|\mathcal{E}| = \frac{a_1 \cdots a_k}{2^k} |B| = \frac{a_1 \cdots a_k}{\Gamma\left(\frac{k+2}{2}\right)} \left(\frac{\sqrt{\pi}}{2}\right)^k .$$

Let S be a set as in the previous lemma, and let $\mathcal{E}_S$ be John's ellipsoid. Assume that $\mathcal{E}_S$ has principal axes of lengths $a_1, \ldots, a_k$. Let R_S be the smallest rectangle whose axes are aligned with those of $\mathcal{E}_S$ that contains $\mathcal{E}_S$ (so that its volume is $a_1 \times \cdots \times a_k$). Then

$$|R_S| \leq \left(\frac{2}{\sqrt{\pi}}\right)^k \Gamma\left(\frac{k+2}{2}\right) |\text{CH}(S)| .$$

The main result of this section clearly shows why we call the n^α term the perimeter complexity. In higher dimensions, the complexity of range search involves the volumes of all the lower-dimensional "facets" of C.

THEOREM 4.4. *Let $U_1, \ldots, U_n$ be independent and uniform random variables over $[0,1]^2$, used to construct either a 2-d tree or a squarish 2-d tree, and let $\mathcal{R}_n$ be the partition into rectangles. Let Z be uniformly distributed over $[0,1]^2$, independent of the U_i's, and let N_n be the number of rectangles in $\mathcal{R}_n$ that intersect $Z+C$, where C is a convex set. Then there is a universal constant $\gamma > 0$ (not depending upon n, Δ, Δ' or C), such that*

$$\mathbf{E}\{N_n\} \leq \gamma\,(n \text{ area}(C) + n^\alpha \text{ perimeter}(C) + \log n) ,$$

where $\alpha = \frac{\sqrt{17}-3}{2}$ if we used 2-d trees and $\alpha = 1/2$ if we used squarish 2-d trees.

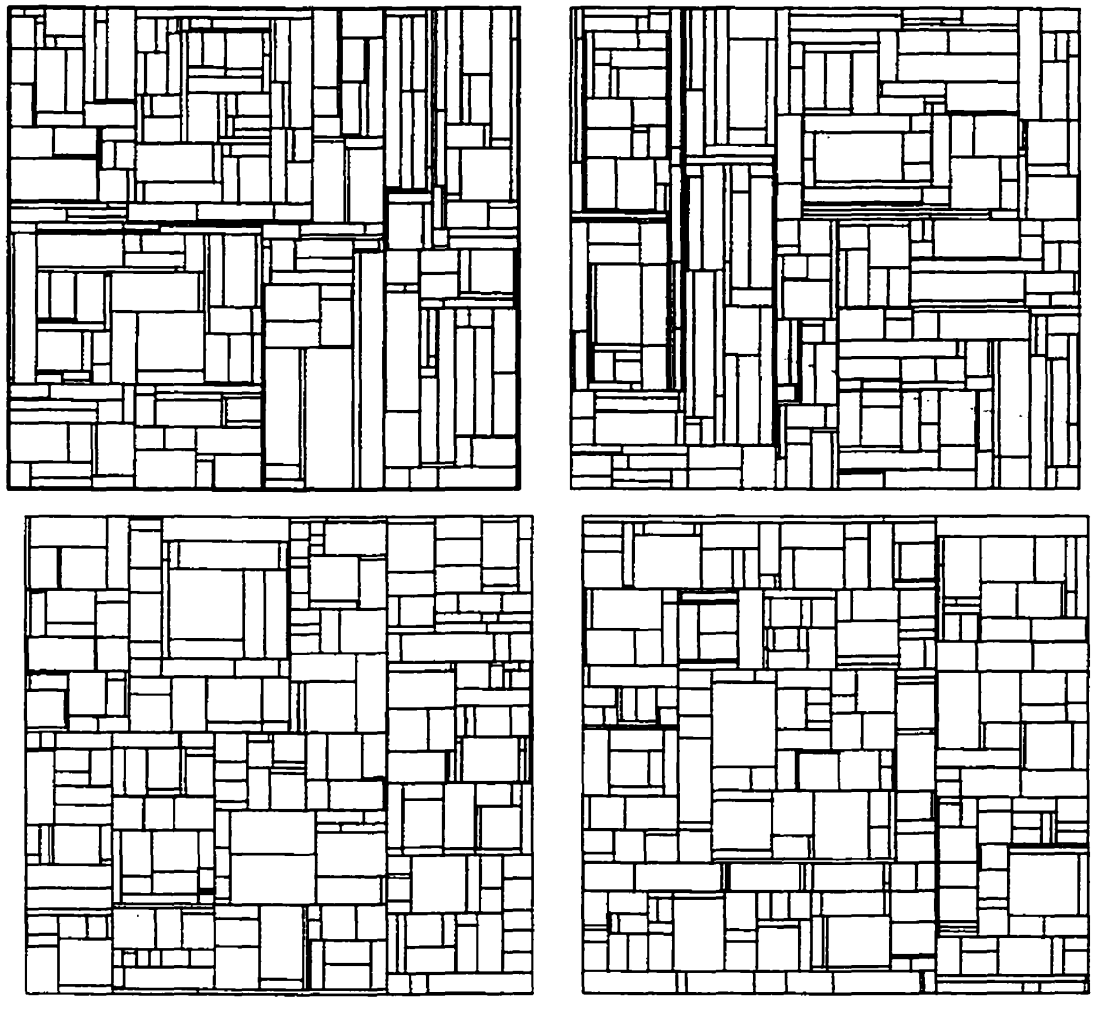

FIGURE 4.5. 2-d and squarish 2-d partitions. The same sets of points are used in both figures. The figures clearly show the elongated nature of rectangles in 2-d trees. The squarish partition looks indeed more "squarish".

PROOF. Let R_C be the rectangle associated to John's ellipsoid $\mathcal{E}_C$ for C, as defined above. Suppose that it is of size $\Delta \times \Delta'$. Note that the number of comparisons that range search performs with $Z + C$ is not more than that for $Z + R_C$. Therefore, for some $\gamma' > 0$,

$$\mathbf{E}\{N_n\} \leq \gamma' \left(n\Delta\Delta' + (\Delta + \Delta')n^\alpha + \log n\right).$$

As we noted earlier, $\Delta\Delta' \leq \frac{4}{\pi}\text{Area}(C)$. By the convexity of C, and using

Also,

$$\text{Perimeter}(R_C) = 2(\Delta + \Delta') \leq 2\sqrt{2}\sqrt{(\Delta)^2 + (\Delta')^2} \leq \sqrt{8}\,\text{Perimeter}(C) \ .$$

Thus we obtain the inequality

$$\mathbf{E}\{N_n\} \leq \gamma' \left((4/\pi) n \text{Area}(C) + \sqrt{8}\,\text{Perimeter}(C) n^{\alpha} + \log n \right) .$$

□

4.4 Conclusions

The reason why range search queries, and in particular partial match queries, have better expected time complexity when using squarish k-d trees than when using k-d trees is because of the more "squarish" nature of the partition on the average that the modified data structure produce. Because k-d trees produce on the average many long skinny rectangles, the probability of a range query region to hit a rectangle in the partition is larger (see figure 4.5).

Chapter 5
Nearest Neighbor Search

In this chapter we propose two algorithms for finding the nearest neighbor among n points. We analyze both algorithms when having as underlying data structure k-d trees and squarish k-d trees.

5.1 Nearest neighbor problem

The nearest neighbor problem is a fundamental problem in areas such as computational geometry and pattern recognition. It has been extensively studied and several techniques have been proposed to solve it, such as branch-and-bound techniques (Fukunaga and Narendra 1975), randomization (Rabin 1976, Yuval 1976), divide-and-conquer (Bentley 1975) and bucketing (Bentley, Weide and Yao 1980). If the data set is of size n, in a worst case scenario in dimension 2 for example, the nearest neighbor problem can be solved optimally by using the Voronoi diagram of the data set in $O(\log n)$ time with $O(n \log n)$ preprocessing time. Here we locate the Voronoi region where the query point is contained, by using binary search on each of the axes. The ancestry of this method can be found in Dobkin and Lipton (1976) (see also Shamos and Hoey, 1975). By the use of hashing techniques, in a probabilistic setting, Bentley, Weide and Yao (1980), showed that the nearest neighbor problem can be solved in $O(1)$ expected time for some distribution of the data. They partition the unit square in an array of size $\sqrt{n}/C$ by $\sqrt{n}/C$ and search the buckets around the bucket to which the query point belongs in a spiral fashion.

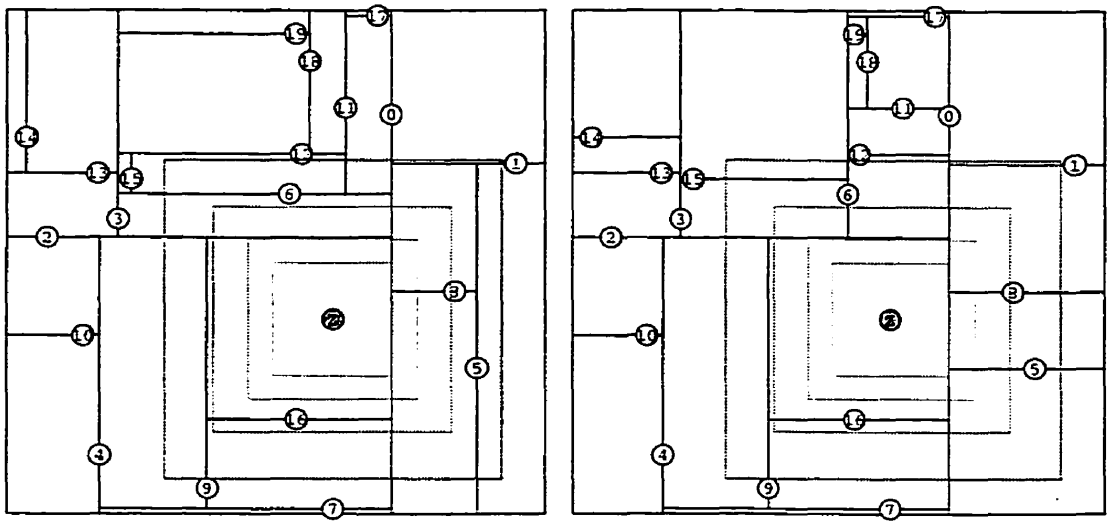

FIGURE 5.1. Illustration of algorithm **A**. The same points are shown in both figures, but to the left a 2-d tree partition and to the right a squarish 2-d tree partition are shown.

We use k-d trees and squarish k-d trees to solve the nearest neighbor problem. We propose two algorithms that use the range search algorithm in the previous chapter, and analyze their expected time complexity.

5.2 Algorithms

To adequately present the results in this section, let us define the following quantity. Given $k \geq 2$, we define

$$\rho_k = \max_{0 \leq s \leq k} \left(\alpha(s/k) - 1 + s/k \right),$$

where $\alpha(\cdot)$ is the Flajolet-Puech function defined in chapter 3. We propose two algorithms for solving the nearest neighbor problem. These are as follows:

- In **algorithm A**, we start with an orthogonal range search with a square box of size $1/n^{1/k}$ centered at the query point Z. Repeat with boxes Q_t of sizes $k^{t/2}/n^{1/k}$ for $t = 1, 2, 3, \ldots$ until $t^* + 1$, where t^* is the index of the first nonempty box. Report the nearest point in the $t^* + 1$-st box (see figure 5.1).

- In **algorithm B**, we insert X in the k-d tree, and let Q be the rectangle associated with X. Let X' be the parent of X in the tree (note: $X' \in Q$). Perform an orthogonal range search centered at X with dimensions $2\|X' - X\|$ in all directions. Report the nearest neighbor among all points returned by this orthogonal range search.

The performance of the two previous algorithms differs depending on which brand of k-d tree is being used. When using algorithm **A** on squarish k-d trees, we prove that the expected complexity time of algorithm **A** is $O(\log n \log\log n)$. If we use algorithm **B** with squarish k-d trees in dimension 2, we prove that the expected complexity time is $O(\log^2 n)$.

5.3 Algorithm A when using k-d trees

Consider a random k-d tree constructed from the insertion of $U_1, \ldots, U_n$, independent uniform random variables over $[0,1]^k$. Let X be a query point uniformly distributed in the unit square. We consider algorithm **A**. The purpose of this section is to prove that the expected complexity of this algorithm is $\Theta(n^{\rho_k})$, where

$$\rho_k = \max_{0 \le s \le k} \theta(s/k).$$

The constant $\rho_k \in (0.061, 0.064)$ depends upon k only, and is $(\sqrt{17} - 4)/2 \approx 0.0615536$ for $k = 2$, is minimal for $k = 3$ ($\rho_k \approx 0.0615254$), and oscillates from that point on. For example, nearest neighbor search in dimensions 2, 4 and 6 have the same expected complexity (as a function of n—the constants may be different), and nearest neighbor search in 3-d is slightly easier than in any other dimension as its ρ_k-value is smallest. The maximal value for ρ_k never exceeds 0.064 (see figure 5.2).

We set first the notation we will use. Let $t \ge 1$, we set for all $1 \le j \le k$, $\Delta_j = k^{t/2}/n^{1/k}$. Let Q_t be the hypercube with sides all equal to $k^{t/2}/n^{1/k}$, centered at X, a random vector uniformly distributed in $[0,1]^k$, on which an orthogonal range search is performed. Let N_t be the number of data points

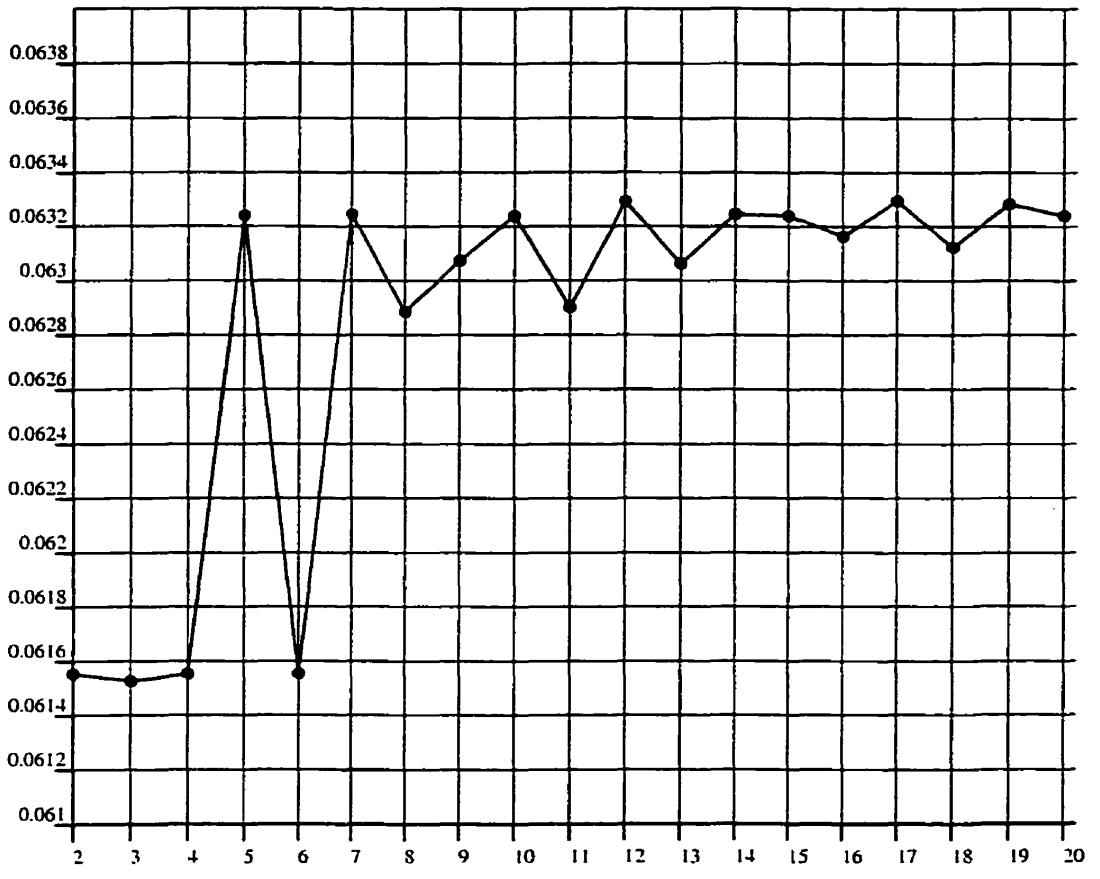

FIGURE 5.2. The function ρ_k versus k, the dimension. The expected complexity of a natural nearest neighbor algorithm grows as n^{ρ_k}.

among $U_1, \ldots, U_n$ falling in Q_t. Let T_t be the complexity of Bentley's orthogonal range search algorithm on Q_t, so that

$$T_t = \sum_{i=1}^{2n+1} 1_{[R_i \cap Q_t \neq \emptyset]} ,$$

where R_i is the rectangle in the partition determined by $U_1, \ldots, U_{i-1}$ in which U_i falls. The time taken by algorithm A is

$$T = T_1 + T_2 + \sum_{t \geq 3} T_t 1_{[N_{t-2}=0]} .$$

We note that by assumption all points fall in the unit hypercube, and therefore, the largest index in the last sum cannot exceed $t^* = \lceil 2 \log n / (k \log k) \rceil$.

FACT. *Let* $\rho_k = \max\{\theta(1/k), \theta(2/k), \ldots, \theta((k-1)/k)\}$. *Then there exists a constant* C *not depending upon* t *or* n *such that*

$$\mathbf{E}\{T_t\} \leq C\left(k^{\frac{(k-1)t}{2}} n^{\rho_k} + k^{\frac{kt}{2}}\right) .$$

Also,

$$\mathbf{P}\{R_i \cap Q_t \neq \emptyset\} \leq \frac{C}{i}\left(k^{\frac{(k-1)t}{2}} i^{\rho_k} + k^{\frac{kt}{2}}\right),$$

for $1 \leq i \leq n$.

PROOF. Using theorem 4.1, with the Δ_j's as previously defined, we obtain

$$\mathbf{E}\{T_t\} \leq$$
$$C\left(k^{\frac{kt}{2}} + k^{\frac{(k-1)t}{2}} n^{\theta(1/k)} + k^{\frac{(k-2)t}{2}} n^{\theta(2/k)} + \cdots + k^{\frac{t}{2}} n^{\theta((k-1)/k)} + \log n\right) .$$

The first inequality in the theorem follows immediately from this and the definition of ρ_k, and the fact that $\log n = o(n^{\rho_k})$. The second inequality uses the fact that $\mathbf{P}\{R_i \cap Q_t \neq \emptyset\}$ is decreasing in i, and thus,

$$i\,\mathbf{P}\{R_i \cap Q_t \neq \emptyset\} \leq \mathbf{P}\{R_1 \cap Q_t \neq \emptyset\} + \cdots + \mathbf{P}\{R_i \cap Q_t \neq \emptyset\} \leq \mathbf{E}\{T_t\},$$

if the sample size used for orthogonal range search is i. The first inequality, with n replaced by i concludes the proof. □

LEMMA 5.1 *Let the following constants be given:* $A > 0$, $\gamma > 0$, $\delta > 0$, $\beta \geq 1$, $1 > \rho > 0$, *subject to the conditions* $A \log \beta \leq 1$, $\log \beta < \delta$. *Then*

$$\sum_{t=1}^{\lceil A \log n \rceil} \beta^t \sum_{i=1}^{n} i^{\rho-1} e^{-\gamma(1-i/n)e^{\delta t}} = O(n^{\rho}) .$$

If the conditions are altered so that $\rho = 0$ *and* $\delta = \log \beta$, *then*

$$\sum_{t=1}^{\lceil A \log n \rceil} \beta^t \sum_{i=1}^{n} \frac{1}{i} e^{-\gamma(1-i/n)e^{\delta t}} = O(\log n) .$$

PROOF. We may assume without loss of generality that $A \log n$ is integer-valued. Consider first the sum

$$\sum_{t=1}^{\infty} \beta^t e^{-\eta e^{\delta t}}$$

where η will later be replaced by $\gamma(1-i/n)$. By comparison with an integral, we see that this is not more than

$$\beta \int_0^\infty \beta^x e^{-\eta e^{\delta x}}\, dx\ .$$

Set $z = \eta e^{x\delta}$, and verify that the latter expression is smaller than

$$\frac{\beta}{\delta\eta}\int_0^\infty (z/\eta)^{\log\beta/\delta - 1} e^{-z}\, dz \le \frac{\beta\Gamma(\log\beta/\delta)}{\delta\eta^{\log\beta/\delta}},$$

for $i < n$. With this inequality in hand, we note that for the n^{th} term in our sum we have

$$\sum_{t=1}^{A\log n} \beta^t n^{\rho-1} \le \frac{n^{\rho-1}n^{A\log\beta}}{1-\frac{1}{\beta}} \le \frac{n^\rho}{1-\frac{1}{\beta}}\ .$$

Furthermore,

$$\sum_{i=1}^{n-1} i^{\rho-1} \sum_{t=1}^{A\log n} \beta^t e^{-\gamma(1-i/n)e^{\delta t}} \le \sum_{i=1}^{n-1} i^{\rho-1} \frac{\beta\Gamma(\log\beta/\delta)}{\delta(\gamma(1-i/n))^{\log\beta/\delta}}$$

and thus, it suffices to show that $\sum_{i=1}^{n-1} i^{\rho-1}(1-i/n)^{-b} = O(n^\rho)$, where $b \in [0,1)$. By comparison with an integral, we have

$$\begin{aligned}\sum_{i=1}^{n-1} i^{\rho-1}(1-i/n)^{-b} &= n^\rho \frac{1}{n}\sum_{i=1}^{n-1}(i/n)^{\rho-1}(1-i/n)^{-b} \\ &\le n^\rho \int_0^1 x^{\rho-1}(1-x)^{-b}\, dx \\ &\le B(\rho, 1-b)n^\rho\ ,\end{aligned}$$

where $B(\cdot,\cdot)$ is the beta integral. This concludes the first part of lemma 5.1. For the second part, note as before that the contributions in the double sum corresponding to $i = n$ and $i = n-1$ are $O(1)$. For the remainder, we have

$$\begin{aligned}&\sum_{t=1}^{A\log n}\sum_{i=1}^{n-2} i^{-1}\beta^t e^{-\gamma(1-i/n)e^{\delta t}} \\ &\quad\le \sum_{t=1}^{\infty}\beta^t e^{-\gamma(1-1/n)e^{\delta t}} + \sum_{t=1}^{A\log n}\int_1^{n-2}\frac{\beta^t}{x}e^{-\gamma(1-(x+1)/n)e^{\delta t}}\, dx \\ &\quad\le O(1) + \frac{\beta}{\delta\gamma}\int_1^{n-2}\frac{1}{x(1-(x+1)/n)}\, dx \\ &\quad= O(\log n)\ .\end{aligned}$$

This concludes the proof of the second part of lemma 5.1. □

THEOREM 5.1. *If T is the time for a nearest neighbor search for algorithm A when using as underlying data structure k-d trees, then* $\mathbf{E}\{T\} = \Theta(n^{\rho_k})$.

PROOF. For the lower bound, we note that $T \geq T_1$, and conclude by the lower bound of theorem 4.1 applied to Q_1 and the definition of ρ_k. For the upper bound, we begin with

$$T = T_1 + T_2 + \sum_{t \geq 3} T_t 1_{[N_{t-2}=0]} \cdot$$

Taking expected values, theorem 4.1 implies that $\mathbf{E}\{T_1 + T_2\} = O(n^{\rho_k})$. We fix $t \geq 3$ and bound $\mathbf{E}\left\{T_t 1_{[N_{t-2}=0]}\right\}$. The two factors in the expected value are dependent. However, if $N_{t-2,i}$ denotes the number of points among $U_{i+1}, \ldots, U_n$ that fall in Q_{t-2}, then we note that given X, $N_{t-2,i}$ and $[R_i \cap Q_t \neq \emptyset]$ are independent. Now note that

$$\begin{aligned}
\mathbf{P}\{N_{t-2,i} = 0|X\} &\leq \sup_{x \in [0,1]^k} \mathbf{P}\{N_{t-2,i} = 0|X = x\} \\
&\leq \left(1 - \left(\frac{\Delta_{t-2}}{2}\right)^k\right)^{n-i} \\
&\leq \exp\left(-\frac{(n-i)k^{\frac{k(t-2)}{2}}}{2^k n}\right) \cdot
\end{aligned}$$

Thus, as $N_{t-2} \geq N_{t-2,i}$, we have by the previous fact

$$\begin{aligned}
\mathbf{E}\left\{T_t 1_{[N_{t-2}=0]}\right\} &= \mathbf{E}\left\{\sum_{i=1}^{2n+1} 1_{[R_i \cap Q_t \neq \emptyset]} 1_{[N_{t-2}=0]}\right\} \\
&\leq 2\,\mathbf{E}\left\{\sum_{i=1}^{n} 1_{[R_i \cap Q_t \neq \emptyset]} 1_{[N_{t-2,i}=0]}\right\} \\
&= 2\,\mathbf{E}\left\{\sum_{i=1}^{n} \mathbf{P}\{R_i \cap Q_t \neq \emptyset|X\}\,\mathbf{P}\{N_{t-2,i} = 0|X\}\right\} \\
&\leq 2\sum_{i=1}^{n} \exp\left(-\frac{(n-i)k^{\frac{k(t-2)}{2}}}{2^k n}\right) \mathbf{E}\{\mathbf{P}\{R_i \cap Q_t \neq \emptyset|X\}\} \\
&= 2\sum_{i=1}^{n} \exp\left(-\frac{(n-i)k^{\frac{k(t-2)}{2}}}{2^k n}\right) \mathbf{P}\{R_i \cap Q_t \neq \emptyset\} \\
&\leq \sum_{i=1}^{n} \exp\left(-\frac{(1-i/n)k^{\frac{k(t-2)}{2}}}{2^k}\right) \frac{2C}{i}\left(k^{\frac{(k-1)t}{2}} i^{\rho_k} + k^{\frac{kt}{2}}\right) \cdot
\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{E}\{T\} &\le O(n^{\rho_k}) + \sum_{t=3}^{t^*} 2Ck^{\frac{(k-1)t}{2}} \sum_{i=1}^{n} \exp\left(-\frac{(1-i/n)k^{\frac{k(t-2)}{2}}}{2^k}\right) i^{\rho_k - 1} \\ &\quad + \sum_{t=3}^{t^*} 2Ck^{\frac{kt}{2}} \sum_{i=1}^{n} \frac{1}{i} \exp\left(-\frac{(1-i/n)k^{\frac{k(t-2)}{2}}}{2^k}\right) \\ &= O(n^{\rho_k}) + I + II\ .\end{aligned}$$

Lemma 5.1 applies to I if we formally take there $\beta = k^{(k-1)/2}$, $\gamma = 1/(2k)^k$, $A = 2/(k\log k)$, and $\delta = (k\log k)/2$. The conditions of the the first part of lemma 5.1, $A\log\beta \le 1$ and $\log\beta < \delta$, hold, so that $I = O(n^{\rho_k})$. The last part of lemma 5.1 applies to II if we set $\beta = k^{k/2}$, $\gamma = 1/(2k)^k$, $A = 2/(k\log k)$, and $\delta = (k\log k)/2 = \log\beta$. Therefore, $II = O(\log n)$. This concludes the proof. □

5.4 Algorithm A when using squarish k-d trees

We analyze first algorithm **A** when using squarish k-d trees. By theorem 4.2, each orthogonal range search taken individually (for fixed i) takes expected time $O(\log n)$. We show in fact that the total expected time is $O(\log n \log\log n)$.

THEOREM 5.2. *Let X be a point uniformly distributed on $[0,1]^k$. Consider a squarish k-d tree based on n i.i.d. points on $[0,1]^k$. Then the expected time of algorithm* **A** *is $O(\log n \log\log n)$.*

PROOF. Let $\mathcal{T}$ be the total time it takes algorithm **A** to finish. Let $\mathcal{T}_i$ be the running time of Bentley's range search algorithm on n i.i.d. points on $[0,1]^k$ and a cube Q_i centered at X of length $k^{t/2}/n^{1/k}$, and let N_t be the number of points in Q_t. Note that

$$\mathbf{E}\{\mathcal{T}\} \le O(\log n) + \mathbf{E}\left\{\mathcal{T}_1 + \mathcal{T}_2 + \sum_{t=3}^{m} \mathcal{T}_t \mathbf{1}_{[N_{t-2}=0]}\right\},$$

where $m = \lfloor \frac{2}{k}\log_k(2^k n)\rfloor$ bounds the maximum number of iterations the

algorithms can perform. Thus, it is enough to prove that

$$\mathbf{E}\left\{\sum_{t=3}^{m} \mathcal{T}_t 1_{[N_{t-2}=0]}\right\} = O(\log n \log\log n).$$

Let $h = \lceil \frac{2}{k} \log_k(2^k \log n) \rceil$, then

$$\mathbf{E}\left\{\sum_{t=3}^{m} T_t 1_{[N_{t-2}=0]}\right\} \leq (h+1)\,\mathbf{E}\left\{\mathcal{T}_{h+1}\right\} + 2n \sum_{t=h+2}^{m} \mathbf{P}\left\{N_{t-2}=0\right\}.$$

Now, by theorem 3.2.

$$\begin{aligned}
&(h+1)\,\mathbf{E}\left\{\mathcal{T}_{h+1}\right\} \\
&\leq \gamma\left(\frac{2}{k}\log_k(2^k\log n)+2\right)\left(k^{(h+1)k/2} + \sum_{\ell=1}^{k-1} n^{1-\ell/k} \sum_{\substack{I\subseteq\{1,\ldots,k\}\\ |I|=\ell}} \prod_{j\notin I} \frac{k^{(h+1)/2}}{n^{1/k}} + \log n\right) \\
&= \gamma\left(\frac{2}{k}\log_k(2^k\log n)+2\right)\left(k^{(h+1)k/2} + \sum_{\ell=1}^{k-1} n^{1-\ell/k}\binom{k}{\ell}\frac{k^{(h+1)(k-\ell)/2}}{n^{1-\ell/k}} + \log n\right) \\
&= \gamma\left(\frac{2}{k}\log_k(2^k\log n)+2\right)\left(k^{(h+1)k/2} + k^{(h+1)k/2}\sum_{\ell=1}^{k-1}\binom{k}{\ell}k^{-(h+1)\ell/2} + \log n\right) \\
&\leq \gamma\left(\frac{2}{k}\log_k(2^k\log n)+2\right)\left(k^{(h+1)k/2}\left(k^{-(h+1)/2}+1\right)^k + \log n\right) \\
&\leq \gamma\left(\frac{2}{k}\log_k(2^k\log n)+2\right)\left(k^k 2^k \log n\left(\frac{1}{\sqrt{k}(2^k\log n)^{1/k}}+1\right)^k + \log n\right) \\
&= O(\log n \log\log n),
\end{aligned}$$

for all $n \geq e$. Finally, for $t \leq m$,

$$\mathbf{P}\left\{N_{t-2}=0\right\} \leq \left(1 - \frac{k^{k(t-2)/2}}{2^k n}\right)^n \leq e^{-k^{k(t-2)/2}/2^k},$$

and therefore $\mathbf{P}\left\{N_{h+2}=0\right\} \leq 1/n$. Thus,

$$2n \sum_{t=h+2}^{m} \mathbf{P}\left\{N_{t-2}=0\right\} \leq 2m = O(\log n).$$

□

Theorem 5.2 is in contrast with the situation presented in the previous section, where for standard random k-d trees, algorithm **A** is shown to take expected time $\Theta(n^{\rho_k})$, where $\rho_k \in (0.061, 0.064)$ depends upon k only.

5.5 Algorithm B when using squarish k-d trees

We analyze now algorithm **B** in dimension 2 only when using squarish 2-d trees as underlying data structure. This algorithm uses the following fact easily proven by induction.

FACT. *Consider the rectangles generated by the insertion of* $x_1, \ldots, x_n \in [0,1]^2$, *in the unit square. Then, for every rectangle* R *in the final partition there is* x_j *lying on the border of* R.

THEOREM 5.3. *Let* X *be a point uniformly distributed on* $[0,1]^2$. *Consider a squarish 2-d tree based on* n *i.i.d. points on* $[0,1]^2$. *Then the expected time of algorithm B is* $O(\log^2 n)$.

We prove a couple of lemmas from which theorem 5.3 follow.

LEMMA 5.2 *Let* $Z, U_1, \ldots, U_n$ *be independent and uniformly distributed random variables on* $[0,1]^2$. *Let* $X_n(Z)$ *and* $Y_n(Z)$ *be the x-length and y-length of the rectangle in the final partition (of the squarish 2-d tree) induced by* $U_1, \ldots, U_n$ *in which* Z *falls. Then, both* $n\,\mathbf{E}\left\{X_n^2(Z)\right\}$ *and* $n\,\mathbf{E}\left\{Y_n^2(Z)\right\}$ *are* $O(\log^2 n)$.

PROOF. By lemmas 3.3 and 3.4, for any $p, q > 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$, we have that

$$\begin{aligned}
\mathbf{E}\left\{X_n^2(Z)\right\} &= \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i^3 Y_i\right\} \\
&\leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} (X_i Y_i)^p\right\}^{1/p} \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i^{2q}\right\}^{1/q} \\
&\leq \left(\frac{4\Gamma(p+1)}{n^{p-1}}\right)^{1/p} \left(\frac{5\Gamma(q+1)}{q-1}\left(q - \frac{1}{n^{q-1}}\right)\right)^{1/q} \\
&= \frac{4^{1/p} 5^{1/q} \left(\Gamma(p+1)\right)^{1/p} \left(\Gamma(q+1)\right)^{1/q}}{(q-1)^{1/q}} \frac{(qn^{q-1}-1)^{1/q}}{n}.
\end{aligned}$$

Let us choose $q = 1 + \frac{1}{\log n}$, $p = \log n + 1$, and assume $n > e$. As $\Gamma(p+1) \leq \sqrt{2\pi}\left(\frac{p}{e}\right)^p e^{1/12p}$ (see for example, Abramowitz and Stegun, 1970), there is

$c > 0$, such that $(\Gamma(p+1))^{1/p} \leq cp = c(\log n + 1)$, and there is $c' > 0$, such that $(\Gamma(q+1))^{1/q} \leq c'q \leq 4c'$. Furthermore, $(q-1)^{-1/q} = (\log n)^{\frac{\log n}{\log n + 1}} \leq \log n$, and $(qn^{q-1} - 1)^{1/q} \leq 2e - 1$. Therefore $n\,\mathbf{E}\left\{X_n^2(Z)\right\} = O(\log^2 n)$. The result for $n\,\mathbf{E}\left\{Y_n^2(Z)\right\}$ follows in the same manner. □

To prove the next lemma we need the following result.

LEMMA 5.3 (DEVROYE, 1986). *Let H_n be the height of a random binary search tree of size n, then for any integer $k \geq \max\{1, \log n\}$ we have*

$$\mathbf{P}\left\{H_n \geq k\right\} \leq \frac{1}{n}\left(\frac{2e\log n}{k}\right)^k.$$

LEMMA 5.4 *Let $Z, U_1, \ldots, U_n$ be independent and uniformly distributed random variables over $[0,1]^2$. Let $X_n(Z)$ and $Y_n(Z)$ be the x-length and y-length of the rectangle in the final partition induced by $U_1, \ldots, U_n$ in which Z falls. Then $\mathbf{E}\left\{X_n(Z)\sum_{i=1}^{2n} X_i\right\}$, $\mathbf{E}\left\{Y_n(Z)\sum_{i=1}^{2n} Y_i\right\}$, $\mathbf{E}\left\{X_n(Z)\sum_{i=1}^{2n} Y_i\right\}$, and $\mathbf{E}\left\{Y_n(Z)\sum_{i=1}^{2n} X_i\right\}$ are $O(\log^2 n)$.*

PROOF. Let $\mathcal{F}_n$ denote the collection of final rectangles in the squarish 2-d tree T constructed from $U_1, \ldots, U_n$. For a final rectangle R_i, denote by $D(R_i)$ its depth. Then $\sum_{i=1}^{2n} X_i \leq \sum_{i\in\mathcal{F}_n} D(R_i)X_i + 1$. Thus if H_n is the height of T,

$$\begin{aligned}
\mathbf{E}\left\{\sum_{i=1}^{2n} X_i X_n(Z)\right\} &\leq \mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} D(R_i)X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} + 1 \\
&\leq \mathbf{E}\left\{H_n \sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} + 1 \\
&\leq t\log n\,\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} + 1 \\
&\quad + \mathbf{E}\left\{1_{[H_n \geq t\log n]} H_n \sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} + 1,
\end{aligned}$$

for any $t > 1$. Using lemma 5.3, we see that,

$$\mathbf{E}\left\{1_{[H_n \geq t\log n]} H_n \sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} \leq n^3\,\mathbf{P}\left\{H_n \geq t\log n\right\}$$
$$\leq n^2 n^{t\log\left(\frac{2e}{t}\right)}.$$

We choose t such that $t\log\left(\frac{2e}{t}\right) < -2$ so that

$$\mathbf{E}\left\{1_{[H_n \geq t\log n]} H_n \sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} = O(1).$$

We complete the proof by showing that

$$\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\} = O(\log n).$$

For this, let $S_r = \sum_{i\in\mathcal{F}_r} X_i \sum_{j\in\mathcal{F}_r} X_j^2 Y_j$, for $r = 1,\ldots,n-1$. Note that

$$S_{r+1} - S_r = \sum_{m\in\mathcal{F}_r} X_m Y_m \left[1_{[X_m<Y_m]} X_m \sum_{j\in\mathcal{F}_r} X_j^2 Y_j\right.$$
$$\left. + 1_{[X_m>Y_m]}((XX_m)^2 Y_m + ((1-X)X_m)^2 Y_m - X_m^2 Y_m)\sum_{i\in\mathcal{F}_r} X_i\right],$$

where $X \stackrel{\mathcal{L}}{=} \text{Uniform}[0,1]$, and is independent of all $U_1,\ldots,U_n$. Now, as $(XX_m)^2 Y_m + ((1-X)X_m)^2 Y_m - X_m^2 Y_m \leq 0$, we have that

$$S_{r+1} - S_r \leq \sum_{i\in\mathcal{F}_r} (X_i Y_i)^{3/2} \sum_{j\in\mathcal{F}_r} X_j^2 Y_j.$$

Note that for any $p, q > 1$, such that $\frac{1}{p} + \frac{1}{q} = 1$,

$$\mathbf{E}\left\{S_{r+1} - S_r\right\} \leq \mathbf{E}\left\{\left(\sum_{i\in\mathcal{F}_r} (X_i Y_i)^{3/2}\right)^p\right\}^{1/p} \mathbf{E}\left\{\left(\sum_{j\in\mathcal{F}_r} X_j^2 Y_j\right)^q\right\}^{1/q},$$

and again by Hölder's inequality, and lemma 3.3, by choosing $q = \sqrt{1.4}$, and $p = \frac{\sqrt{1.4}}{\sqrt{1.4}-1}$,

$$\mathbf{E}\left\{\left(\sum_{i\in\mathcal{F}_r} (X_i Y_i)^{3/2}\right)^p\right\}^{1/p} \leq \mathbf{E}\left\{r^{p/q}\sum_{i\in\mathcal{F}_r} (X_i Y_i)^{3p/2}\right\}^{1/p} \leq \frac{12}{\sqrt{r}}.$$

By applying Hölder's inequality inside the expected value,

$$\begin{aligned}
\mathbf{E}\left\{\left(\sum_{j\in\mathcal{F}_r} X_j^2 Y_j\right)^q\right\}^{1/q} &\le \mathbf{E}\left\{r^{q/p}\sum_{j\in\mathcal{F}_r}(X_j^2Y_j)^q\right\}^{1/q} \\
&\le r^{1/p}\left(\mathbf{E}\left\{\sum_{j\in\mathcal{F}_r}(X_jY_j)^{qp}\right\}^{1/p}\mathbf{E}\left\{\sum_{j\in\mathcal{F}_r}X_j^{q^2}\right\}^{1/q}\right)^{1/q} \\
&\le 46\, r^{1/p}\left(\left(\frac{1}{r^{qp-1}}\right)^{1/p}\left(\frac{1}{r^{q^2/2-1}}\right)^{1/q}\right)^{1/q} \\
&= \frac{46}{\sqrt{r}}\,.
\end{aligned}$$

Thus, $\mathbf{E}\{S_{r+1}-S_r\} \le 552/r$, and by summing the differences we finally can conclude that $\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n} X_i \sum_{j\in\mathcal{F}_n} X_j^2 Y_j\right\}$ is indeed $O(\log n)$. The other expected values can be bounded in the same way. □

PROOF OF THEOREM 5.3. Given $U_1,\ldots,U_n$, we define $L_n(Z) = 2(X_n(Z) + Y_n(Z))$. Note that as the expected height of T is $O(\log n)$, the expected time complexity of the nearest neighbor algorithm is bounded by $O(\log n)$ plus the expected time of random orthogonal range search with query rectangle Q having all sides of length $L_n(Z)$, and centered at Z. Let N_n be the time complexity of range search. By the same arguments followed in theorem 4.3 we have,

$$\mathbf{E}\{N_n\} \le \mathbf{E}\left\{\sum_{i=1}^{2n+1} X_iY_i\right\} + 2\,\mathbf{E}\left\{\sum_{i=1}^{2n+1} L_n(Z)(X_i+Y_i)\right\} + 8n\,\mathbf{E}\{L_n^2(Z)\} + 1.$$

By lemma 3.4, $\mathbf{E}\left\{\sum_{i=1}^{2n+1} X_iY_i\right\} = O(\log n)$. For $\mathbf{E}\left\{\sum_{i=1}^{2n+1} L_n(Z)(X_i+Y_i)\right\}$, lemma 5.4 above shows that it is $O(\log^2 n)$. As we have that

$$n\,\mathbf{E}\{X_n(Z)Y_n(Z)\} = n\,\mathbf{E}\left\{\sum_{i\in\mathcal{F}_n}(X_iY_i)^2\right\},$$

lemma 3.3 shows that it is $O(1)$. Finally, by lemma 5.2 we have that $n\,\mathbf{E}\{L_n^2(Z)\} = O(\log^2 n)$. Thus the expected running time of algorithm B is $O(\log^2 n)$. □

5.6 Lower bound for nearest neighbor queries

Friedman, Bentley and Finkel (1975) defined an optimized k-d tree on which associative queries take optimal expected time. They defined optimized k-d trees, so that at every node the coordinate with the largest spread in values is chosen as the discriminator and the median of the discriminator key values partitions the space. The time to construct the tree requires that at each level of the tree the entire set of keys be scanned. This requires computation of $O(kn)$, for n records. As the depth of the tree is $O(\log n)$, the total computation to construct the tree is $O(kn \log n)$. They propose a recursive algorithm for solving nearest neighbor queries that works as follows. The algorithm starts by computing the distance d between the root and the query point, this yields an estimated nearest neighbor distance. The ball B centered at the query point and radius d is considered. The search continues recursively along the left and/or right subtrees according to whether the B intersects the rectangles associated to the left and right subtrees updating d in each recursive invocation. They experimentally investigated the running time of this algorithm and observed that off-line nearest neighbor queries may be solved in $O(\log n)$ expected time using optimized k-d trees. Later, Bentley (1990) proposed to study top-down and bottom-up off-line nearest neighbor queries using optimized k-d trees. In a top-down nearest neighbor query we descend the optimized k-d tree to find the data point in the tree, perform a nearest neighbor search down the node's subtree and go up the tree whenever there is a chance there may be a closer node outside the tree rooted at the query point. In a bottom-up nearest neighbor search we assume that we are already at the node associated to the query point and perform a nearest neighbor search. Bentley (1990) experimentally studied and conjectured that top-down and bottom-up nearest neighbor queries can be solved off-line in $O(\log n)$ and $O(1)$ expected complexity time when using optimized k-d trees.

Note that if the data are put in a $\sqrt{n} \times \sqrt{n}$ regular grid partition of $[0,1]^2$, then each cell would receive on average one data point. Bentley, Weide and Yao (1980) showed that nearest neighboring searching starting from a given point in a cell takes $O(1)$ expected time. The same is true for all sufficiently regular, dense and rotund partitions, including, for example, the Voronoi diagram or the Delaunay triangulation. If the data are stored in a 2-d tree however, the property fails to hold because of the skinny rectangles. To see intuitively what is going on, let X be U_1 and let X' be the nearest neighbor of X among $U_2, \ldots, U_n$. Define the nearest neighbor distance $D_n = \|X - X'\|$. Note that D_n is $\Theta(1/\sqrt{n})$ in probability, i.e., $\mathbf{P}\{D_n = o(1/\sqrt{n})\} = o(1)$ and $\mathbf{P}\{D_n = \omega(1/\sqrt{n})\} = o(1)$. This means that a nearest neighbor search for X is roughly equivalent to a $c/\sqrt{n} \times c/\sqrt{n}$ range search. Indeed, just to verify that X' is in fact the claimed nearest neighbor of X, one must at the very least inspect all nodes on rectangle edges that cut the circle S centered at X with radius D_n. Since the rectangles are skinny, the points on the edges may in fact be far from X. Thus a lower bound on the complexity is

$$\mathbf{E}\left\{\sum_{i=n+1}^{2n+1} 1_{[R_i \cap Q \neq \emptyset]}\right\},$$

where Q is the circle of radius D_n centered at X. As D_n is in probability $\Theta(1/\sqrt{n})$, theorem 4.1 implies that the expected complexity is $\Omega(n^{\alpha(1/2)-1/2}) \geq \Omega(n^{0.0615\ldots})$. Algorithm **A** is a very natural on-line nearest neighbor algorithm. In this chapter we showed that its expected complexity time is $\Theta(n^{\rho_k})$ when using k-d trees. We conjecture that indeed any algorithm that computes the nearest neighbor using k-d trees as data structure for storing the data must have expected time complexity $\Omega(n^{\rho_k})$.

5.7 Conclusions

The bound for algorithm **B** is a bit worse than that for algorithm **A**, because while most rectangles are squarish, a sufficient number of them are

elongated. In fact, for given $M > 1$, about $1/M$ of the final (leaf) rectangles or more should have an edge ratio exceeding M. For edge ratio M, and considering that all rectangle areas are about $1/n$, we see that the orthogonal range search should take about M points (the longest edge is about $\sqrt{M/n}$). The expected number of returned elements is at least $\Theta(\log n)$. And the expected number of leaf rectangles visited is of the same order. But each visited leaf rectangle also induces a visit to all of its ancestors, and there are about $\log n$ of those. The proof of the bound for algorithm **A** on squarish k-d trees is by no means optimal. We believe that its real expected time complexity is $\Theta(\log n)$. Note that $\Omega(\log n)$ is an almost trivial lower bound for algorithm **A** on squarish k-d trees.

PART II

Branch-and-Bound Search

Chapter 6
Branching processes

In this chapter we introduce the main definitions and recall or prove auxiliary results for the next chapter, in which we analyze the complexity of branch-and-bound search on random b-ary trees.

6.1 Definitions and basic properties

Around 1874 Galton and Watson introduced a model for studying the "problem of extinction of families" in England. Although their process hardly applies to their original problem, it has become a powerful tool for analyzing different phenomena in areas such as biology, physics and computer science.

We can visualize a branching process as a possible infinite tree. The root has Z_1 children, where Z_1 has a fixed distribution $(p_i)_{i\geq 0}$ (the reproduction distribution). Each child in turn reproduces independently according to the reproduction distribution. This leads to the Galton-Watson random tree, and the Galton-Watson process. We denote by Z_i the number of children in the i^{th} generation in the Galton-Watson tree, with $Z_0 = 1$. We introduce now the RGF (reproduction generating function),

$$f(s) = \sum_{t=0}^{\infty} p_t s^t = \mathbf{E}\left\{s^{Z_1}\right\}, \quad s \in [0,1].$$

The reproduction generating function is a very convenient tool for analyzing the behavior of Galton-Watson branching processes. Let us define the Malthusian parameter which is nothing but the expected number of children per particle.

$$m = \mathbf{E}\left\{Z_1\right\} = \sum_{t=0}^{\infty} t p_t = f'(1).$$

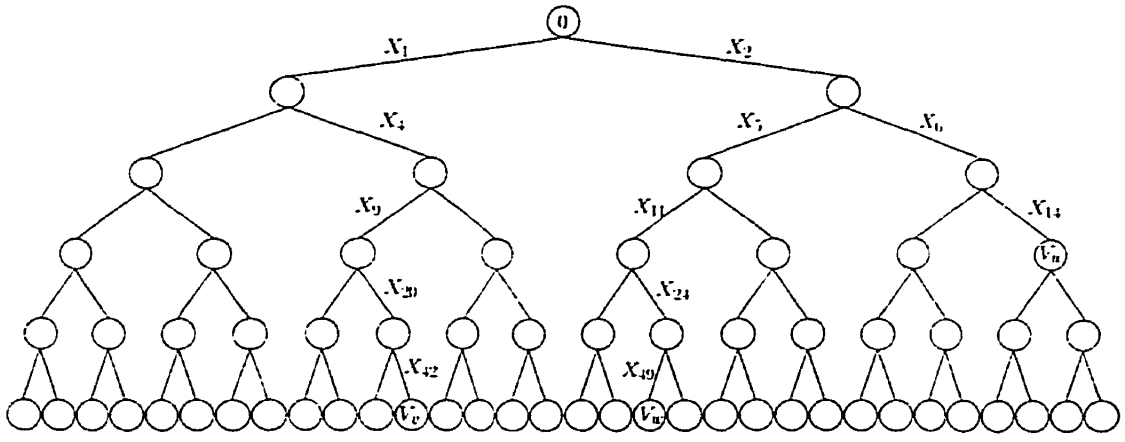

FIGURE 6.1. Two-ary tree showing some edges and node values.

Only two situations can occur: either the population survives forever or it becomes extinct after a finite time. If the expected number of children per particle is greater than one, the population explodes, and if it is less than one the population will die out. Consider the RGF for Z_n, the size of the n^{th} generation:

$$f_n(s) = \mathbf{E}\left\{s^{Z_n}\right\}, \quad s \in [0,1].$$

Let us define q to be the probability that the process becomes extinct. Note that $f_n(0) = \mathbf{P}\{Z_n = 0\}$. In fact, when $m \leq 1$, it can be seen, by manipulating $f_n(s)$, that $q = 1$ (unless the degenerate case $p_1 = 1$ happens), and when $m > 1$, $q < 1$.

6.2 Theory of branching random walks

In a branching random walk, a random walk is superimposed on each path from the root down in a Galton-Watson tree. A value V_v is assigned to each node in the tree, the value of the root being zero. We consider the following type of branching random walk. Given a node u in the Galton-Watson tree, for every child v of u, we define $V_v = V_u + X_v$, with all displacements X_v independent. Note that this is equivalent to assigning to each edge a random variable and assigning to each child of a node the value assigned to

the edge joining them plus the value of the parent.

In general, if u is a node in the tree and its children have displacements $X_{v_1}, \ldots, X_{v_N}$, where N is the size of the offspring of u, then the joint distribution of $(N, X_{v_1}, \ldots, X_{v_N})$ is quite arbitrary. What is important is that each parent produces children (and their values) in the same manner. We assume further that the number of children per parent is a fixed positive integer b (see figure 6.1).

We will use some results adapted from the theory of branching random walks in order to prove the results in this part of the thesis. For additional information see, for example, Asmussen and Hering (1983), Athreya and Ney (1972), and Harris (1963).

We prove our results by looking at the properties of the μ-function, which we define below. For any random variable X, we define $m(\theta) = b\,\mathbf{E}\left\{e^{-\theta X}\right\}$, $\theta \geq 0$. We assume that $m(\theta) < \infty$, for some $\theta \geq 0$.

DEFINITION. *Let $X \geq 0$ be a nondegenerate random variable. For any $a \in \mathbb{R}$, the μ-function is defined by*

$$\mu(a) = \inf_{\theta \geq 0}\{e^{\theta a}\, m(\theta)\} = b \inf_{\theta \geq 0} \mathbf{E}\left\{e^{\theta(a-X)}\right\}.$$

For each $t > 0$, and $n > 0$, we define

$$Z^{(n)}(t) = \sharp\{w : w \text{ is a leaf in } T_n, \text{ and } V_w \leq t\},$$

where $\sharp A$ denotes the cardinality of set A, and T_n is the complete binary tree of height n. Thus $Z^{(n)}(t)$ is the number of individuals in the n^{th} generation of the process with value smaller or equal to t.

The following results are from Kingman (1975) and Biggins (1977).

THEOREM 6.1. *If $\mu(a) < 1$, then with probability one,*

$$Z^{(n)}(na) = 0 \qquad \textit{for all but finitely many } n.$$

If $a \in \text{int}\{a : \mu(a) > 1\}$, *then*

$$(Z^{(n)}(na))^{1/n} \longrightarrow \mu(a) \qquad \textit{almost surely.}$$

THEOREM 6.2. *Let* T *be the infinite* b*-ary random tree having edge values distributed as* $X \geq 0$, *where* X *is nondegenerate. Let* $B_n = \min\{V_v : v \text{ is a leaf of } T_n\}$. *Then,*

$$\lim_{n\to\infty} \frac{B_n}{n} = \alpha \stackrel{\text{def}}{=} \inf\{a : \mu(a) > 1\},$$

almost surely.

We now prove some properties of the μ-function.

THEOREM 6.3. *Let* $X \geq 0$ *be a nondegenerate random variable. Then its* μ*-function satisfies the following properties:*

(1) μ *is an increasing function on* $[0, \infty)$.

(2) μ *is continuous on* $\text{int}\{a : \mu(a) > 0\}$.

(3) $\log \mu$ *is concave.*

(4) $\sup_{a\in\mathbf{R}} \mu(a) \leq b$.

(5) *If* $\mathbf{E}\{X\} < \infty$, *then* $\mu(a) \equiv b$, *on* $\text{int}\{t : \mu(t) > 0\}$, *for* $a \geq \mathbf{E}\{X\}$.

(6) $\lim_{a\uparrow\infty} \mu(a) = b$.

(7) *If* $X \geq c > 0$, *then* $\mu(a) = 0$ *for* $a < c$.

(8) *Let* $s = \sup\{t : \mathbf{P}\{X < t\} = 0\}$, *and define* $p = \mathbf{P}\{X = s\}$. *Then* μ *is continuous on*
(s, ∞), $\mu(s) = bp$, *and* $\mu(a) = 0$ *for* $a < s$.

PROOF. For proofs of (1)-(3), we refer to Biggins (1977) or Kingman (1975).
Proof of (4). Clearly $\inf_{\theta\geq 0} \mathbf{E}\left\{e^{\theta(a-X)}\right\} \leq 1$ by evaluation at $\theta = 0$.
Proof of (5). For $a \geq \mathbf{E}\{X\}$, we have

$$\begin{aligned}
\mu(a) &= b \inf_{\theta\geq 0} \mathbf{E}\left\{e^{\theta(a-X)}\right\} \\
&\geq b \inf_{\theta\geq 0} e^{\theta(a-\mathbf{E}\{X\})} \qquad \text{(by Jensen's inequality)} \\
&\geq b \qquad \text{(as the infimum is attained at } \theta = 0\text{)}.
\end{aligned}$$

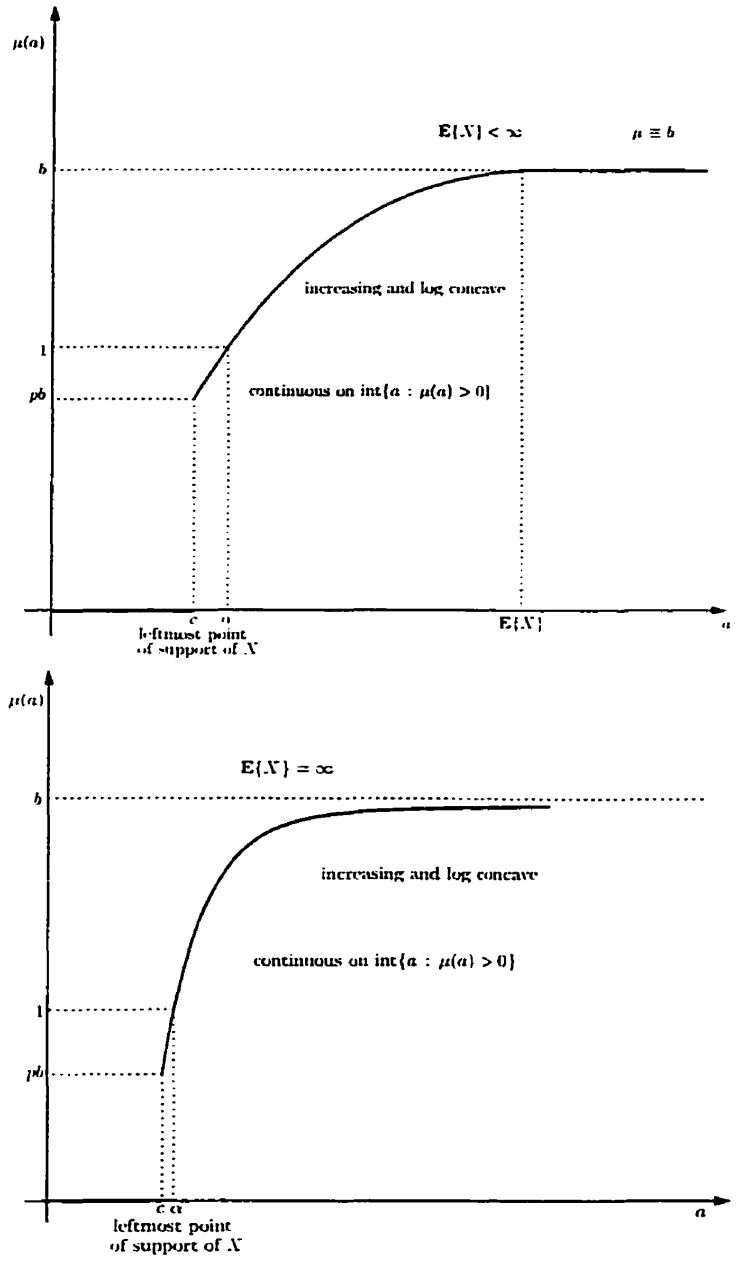

FIGURE 6.2. General form of the μ function.

Proof of (6). If $\mathbf{E}\{X\} < \infty$, this follows from (4) and (5). If $\mathbf{E}\{X\} = \infty$, then let q_{1-p} be the $(1-p)$-th quantile of X, i.e. $q_{1-p} = \sup\{t : \mathbf{P}\{X \leq t\} \leq 1-p\}$. Clearly, as $p \downarrow 0$, $q_{1-p} \uparrow \infty$, since $\mathbf{E}\{X\} = \infty$. Now, for $a \geq q_{1-p}$,

$$\begin{aligned}\mu(a) &\geq b\left(\inf_{\theta \geq 0} e^{\theta(a-q_{1-p})}\right)(1-p) \\ &\geq b(1-p).\end{aligned}$$

We conclude that proof by letting $p \downarrow 0$.

Proof of (7). For $a < c$,

$$\mu(a) = b \inf_{\theta \geq 0} \mathbf{E}\left\{e^{\theta(a-X)}\right\} \leq b \inf_{\theta \geq 0} e^{\theta(a-c)} \leq b \liminf_{\theta \to \infty} e^{\theta(a-c)} = 0.$$

Proof of (8). We only need to show that $\mu(s) = bp$, as the other statements follow from (2) and (7). Define $p + \delta = \mathbf{P}\{s \leq X < s + \varepsilon\}$. Then for all $\varepsilon > 0$ small enough

$$\begin{aligned}
\frac{\mu(s)}{b} &= \inf_{\theta \geq 0} \mathbf{E}\left\{e^{\theta(s-X)}\right\} \\
&\leq \inf_{\theta \geq 0} \left\{e^{\theta(s-s)}(p+\delta) + (1-p-\delta)e^{\theta(s-s-\varepsilon)}\right\} \\
&= \inf_{\theta \geq 0} \left\{p + \delta + (1-p-\delta)e^{-\theta\varepsilon}\right\}.
\end{aligned}$$

But the last expression goes to p as $\varepsilon \to 0$, by taking $\theta = 1/\varepsilon^2$ (notice that $\delta \to 0$ as $\varepsilon \to 0$). Thus $\mu(s) \leq bp$. Also, observe that

$$\frac{\mu(s)}{b} \geq \inf_{\theta \geq 0} pe^{\theta(s-s)} = p.$$

□

We conclude that μ must always follow the pattern as described in figure 6.2.

Let X be a nonnegative and nondegenerate random variable. We say that X is *regular*, if and only if $b\mathbf{P}\{X = c\} < 1$, where c is the leftmost point of the support of X. For regular random variables we define

$$\begin{aligned}
\alpha &= \inf\{a : \mu(a) > 1\}, \\
\beta &= \sup_{x \in (0,1)} \mu^x\left(\frac{\alpha}{x}\right), \\
\gamma &= \inf\left\{y : \mu^y\left(\frac{\alpha}{y}\right) = \beta\right\}.
\end{aligned}$$

Clearly α is well-defined, finite and positive. Also (as we will see below), the solution γ of $\mu^\gamma(\alpha/\gamma) = \beta$ is unique, and $0 < \beta < b$, strictly.

We now prove two additional properties of the μ-function for regular random variables.

LEMMA 6.1 *Let $X \geq 0$ be regular. Then:*

(9) *For all $\varepsilon > 0$, there is $\xi > 0$ such that*

$$\sup_{x \in (0,1)} \left\{ \mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right) \right\} \leq \varepsilon.$$

(10) *For all $\eta > 0$, there is $\xi > 0$ such that for all $0 \leq \nu \leq \xi$,*

$$\sup_{x \in (0,1)} \mu^x\left(\frac{\alpha+\nu}{x}\right) \leq \sup_{x \in (0,1)} \mu^x\left(\frac{\alpha}{x}\right) + \eta.$$

PROOF. Proof of (9). As μ is continuous, bounded, and log-concave on $[\alpha, \infty)$, it is uniformly continuous there, and thus for $\varepsilon > 0$ there is $\delta_\varepsilon > 0$ such that for all $\xi > 0$, small enough,

$$\left| \frac{\alpha+\xi}{x} - \frac{\alpha}{x} \right| = \frac{\xi}{x} \leq \delta_\varepsilon$$

implies that $\mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right) \leq \varepsilon$. If $\xi/x > \delta_\varepsilon$, then $\alpha/x \geq (\alpha/\xi)\delta_\varepsilon$, and by choice of $\xi = \delta_\varepsilon^2$, we see that $\alpha/x \geq \alpha/\delta_\varepsilon$, so that

$$\sup_{\frac{\xi}{x} > \delta_\varepsilon} \left\{ \mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right) \right\} \leq b - \mu\left(\frac{\alpha\delta_\varepsilon}{\xi}\right) = b - \mu\left(\frac{\alpha}{\delta_\varepsilon}\right),$$

which is small enough by the choice of δ_ε.

Proof of (10). By the triangle inequality, we need only to show that

$$\sup_{x \in (0,1)} \left\{ \mu^x\left(\frac{\alpha+\xi}{x}\right) - \mu^x\left(\frac{\alpha}{x}\right) \right\} \leq \eta,$$

for $\xi > 0$ small enough. By $(a+b)^p - a^p \leq pba^{p-1}$, for all $a, b > 0$, $p \in [0,1]$, we have

$$\begin{aligned}
\sup_{x \in (0,1)} \left\{ \mu^x\left(\frac{\alpha+\xi}{x}\right) - \mu^x\left(\frac{\alpha}{x}\right) \right\} &\leq \sup_{x \in (0,1)} \left\{ x\left(\mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right)\right) \mu^{x-1}\left(\frac{\alpha}{x}\right) \right\} \\
&\leq \sup_{x \in (0,1)} \left\{ \mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right) \right\} \sup_{x \in (0,1)} \left\{ x\,\mu^{x-1}\left(\frac{\alpha}{x}\right) \right\} \\
&\leq \sup_{x \in (0,1)} \left\{ \mu\left(\frac{\alpha+\xi}{x}\right) - \mu\left(\frac{\alpha}{x}\right) \right\},
\end{aligned}$$

as $1 \leq \mu(\alpha/x)$, for $x \in (0,1)$. □

LEMMA 6.2 *Let* $s = \sup\{t : \mathbf{P}\{X < t\} = 0\}$. *Assume that* $s < \alpha$. *Then*

$$\beta = \sup_{x \in (0,1)} \mu^x \left(\frac{\alpha}{x}\right) < b.$$

PROOF. By continuity of μ, there is $\varepsilon > 0$, such that $\mu(\alpha/(1-\varepsilon)) \leq \sqrt{b}$. Then, because $s < \alpha$,

$$\begin{aligned}
\beta &= \max\left\{\sup_{0<x\leq 1-\varepsilon} \mu^x \left(\frac{\alpha}{x}\right), \sup_{1-\varepsilon<x<1} \mu^x \left(\frac{\alpha}{x}\right)\right\} \\
&\leq \max\left\{b^{1-\varepsilon}, \mu\left(\frac{\alpha}{1-\varepsilon}\right)\right\} \\
&\leq \max\left\{b^{1-\varepsilon}, \sqrt{b}\right\},
\end{aligned}$$

the result follows. □

Chapter 7
Random b-ary trees

In this chapter we present our main result about the time complexity of branch-and-bound search for random b-ary trees. We use branching random walks as a tool for analyzing the behavior of the algorithm on random b-ary trees.

7.1 Introduction

Let T_n be a b-ary tree of height n, which has independent, nonnegative, identically distributed random variables associated with each of its edges. The value of a node is the sum of all the edge values on its path to the root. We consider the problem of finding the minimum leaf value of T_n. Assume that the edge random variable X is nondegenerate, has $\mathbf{E}\{X^\theta\} < \infty$ for some $\theta > 2$, and satisfies $bP\{X = c\} < 1$ where c is the leftmost point of the support of X. We analyze the performance of the standard branch-and-bound algorithm (this is, the nodes are visited in a depth-first search fashion trimming useless branches) for this problem and prove that the number of nodes visited is in probability $(\beta + o(1))^n$, where $\beta \in (1, b)$ is a constant depending only on the distribution of the edge random variables. We derive explicit expressions for β. We also show that any search algorithm must visit $(\beta + o(1))^n$ nodes with probability tending to one, so branch-and-bound is asymptotically optimal where first-order asymptotics are concerned.

7.2 Previous work

Karp and Pearl (1983) introduced the following model. Let T_n be a binary tree of height n, which has independent, identically distributed Bernoulli(p)

random variables associated with each of its edges. The value of a node is the sum of the values of the edges on the path from the root to that node. The objective is to find the leaf in the tree with minimal value. Karp and Pearl noted that if $2p > 1$, any algorithm must necessarily take exponential time in n, while for $2p = 1$ and $2p < 1$, ordinary uniform cost breadth-first search takes on the average $\Theta(n^2)$ and $\Theta(n)$ time. In this algorithm, one first visits all nodes of value 0, then all nodes of value 1, and so forth.

In 1990, McDiarmid and Provan and McDiarmid (1990) generalized the work of Karp of Pearl to b-ary trees and more general nonnegative edge distributions. If X is a typical edge random variable, and $p = \mathbf{P}\{X = 0\}$, where 0 is the leftmost point of the support of X, they show that if $bp < 1$, any exact search algorithm must take exponential time. It is this model the one we will consider. We assume throughout that X is a regular random variable, as defined in the previous chapter. As the tree T_n has b^n leaves, it is important to ask what fraction of the nodes is revealed before the minimal leaf is found. In branch-and-bound search, we visit the nodes as in depth-first search, and visit v if and only if its parent's value is less than the minimal leaf value seen thus far (if any have been visited; otherwise, the node is visited unconditionally). The algorithm is of course guaranteed to find the overall minimum. If N is the number of nodes visited by the branch-and-bound algorithm (the number of values V_v revealed), we will show the following theorem.

THEOREM 7.1. *Let X be a regular random variable.*

A. *If N is the number of nodes visited by branch-and-bound search, and $\mathbf{E}\{X^\theta\} < \infty$ for some $\theta > 2$, then there exists a number $\beta \in (1, b)$ such that*

$$\lim_{n\to\infty} \mathbf{P}\left\{\left|N^{1/n} - \beta\right| > \epsilon\right\} = 0. \tag{1}$$

B. *The number N' of nodes visited by any algorithm that is guaranteed to find the optimum must be such that*

$$\lim_{n\to\infty} \mathbf{P}\left\{N'^{1/n} \le \beta - \epsilon\right\} = 0$$

for all $\epsilon > 0$.

Sometimes it is instructive to see how N compares with the size of T_n. The theorem above states that in probability,

$$N = |T_n|^{\rho + o(1)}$$

as $n \to \infty$, where $|T_n| = \sum_{i=0}^{n} b^i$ and $\rho = \log_b \beta$. The closer ρ is to zero, the more pruning is achieved. Interestingly, the pruning parameter ρ differs from distribution to distribution, and may take any value in $(0, 1)$. As $\rho > 0$, we see that an exponential explosion (in n) is unavoidable. The proofs are based upon results from branching random walks due to Biggins (1977). Especially part B of theorem 7.1 is an embarrassingly straightforward corollary of Biggins' results. We give an explicit form for β and ρ for all regular distributions.

The same tree model was also considered by Zhang and Korf (1992), who analyzed other search strategies, such as iterative-deepening-A* and recursive best-first search. Branch-and-bound was also analyzed by Smith (1984) on a different random tree model. Wah and Yu (1985) considered branch and-bound with best-first search, and Stone and Sipala (1986) looked at backtracking. Of course, backtracking was analyzed on a host of other models, and we refer to Purdom (1983) and Brown and Purdom (1981) for just two examples. Pearl (1984) is the basic reference for the probabilistic analysis of various search strategies.

7.3 Notation and preliminary results

Let us first set the notation. Let $u_0, \ldots, u_n$ be the nodes in the left roof of T_n, let $v_{k,1}, \ldots, v_{k,b-1}$ be the siblings of u_k, and thus the children of u_{k-1} (see figure 7.1). Let $V_{u,v}$ denote the sum of all edge values on the path from u to v. Notice that $V_{u,v} = V_v - V_u$, if v is a descendent of u. We will be particularly interested in $V_{u_{k-1},w}$, for descendants w of u_{k-1}. We define the

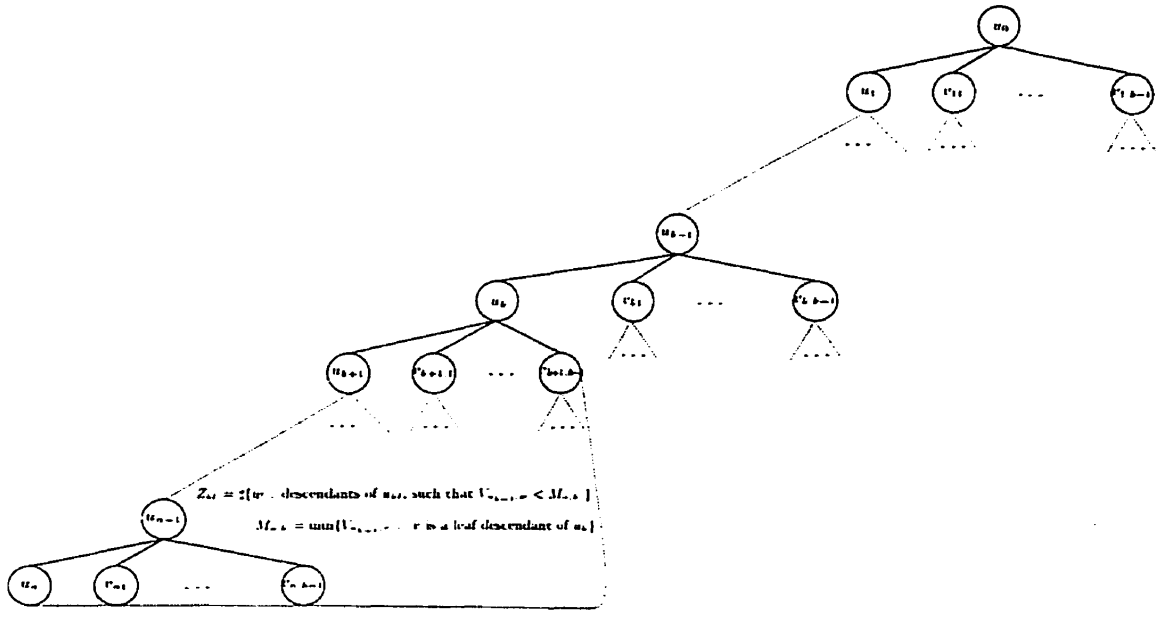

FIGURE 7.1. Notation for trees.

following random variables that will appear in our analysis:

$M_{n,k} =$

$\min\left\{V_{u_{k-1},v} \,:\, v \text{ is a leaf descendant of } u_k, \text{ where } v \text{ and } u_k \text{ are nodes in } T_n\right\},$

$Z_{kl} =$

$\sharp\left\{w \,:\, \text{descendants of } v_{kl} \text{ (including } v_{kl} \text{ itself), such that } V_{u_{k-1},w} < M_{n,k}\right\}$

We now present some preliminary results in order to prove part A of theorem 7.1.

PROPOSITION 7.1 *Assume* $\mathbf{E}\left\{X^\theta\right\} < \infty$ *for some* $\theta > 2$. *Consider* T_n, *and define* $N_k = \max\{V_w \,:\, w \text{ is a descendant of } u_1 \text{ at distance } k \text{ from } u_1\}$, *where* $k = \lfloor d \log_b(n-1) \rfloor$, *for* $0 < d < \theta - 2$ *fixed. Then, for every* $\xi > 0$ *there exists* $\zeta > 0$, $n_0 > 0$, *such that*

$$\mathbf{P}\left\{N_k > (n-1)\xi\right\} \leq \frac{\zeta}{(n-1)^2}, \text{ for all } n \geq n_0.$$

PROOF. Clearly, if $N_k > (n-1)\xi$, there must be at least one node descendant of u_1 at depth $k+1$ having value greater than $(n-1)\xi$. Therefore $\mathbf{P}\left\{N_k > (n-1)\xi\right\} \leq b^k \mathbf{P}\left\{\sum_{i=1}^{k+1} X_i > (n-1)\xi\right\}$, where $X_i \stackrel{\mathcal{L}}{=} X$, for

$i = 1, \ldots, k$, and the X_i's are independent. Then by Markov's and Jensen's inequalities, for all n large enough,

$$\begin{aligned} b^k \mathbf{P}\left\{\sum_{i=1}^{k+1} X_i > (n-1)\xi\right\} &\leq \frac{b^k \mathbf{E}\left\{\left(\sum_{i=1}^{k+1} X_i\right)^\theta\right\}}{((n-1)\xi)^\theta} \\ &\leq \frac{b^k (k+1)^\theta \mathbf{E}\left\{X^\theta\right\}}{((n-1)\xi)^\theta} \\ &\leq \frac{b^{\lfloor d \log_b(n-1)\rfloor} \left(\lfloor d \log_b(n-1)\rfloor + 1\right)^\theta \mathbf{E}\left\{X^\theta\right\}}{((n-1)\xi)^\theta} \\ &\leq \frac{(2d)^\theta \mathbf{E}\left\{X^\theta\right\}}{\xi^\theta} \frac{(\log_b(n-1))^\theta}{(n-1)^{\theta-d}} \\ &\leq \frac{(2d)^\theta \mathbf{E}\left\{X^\theta\right\}}{\xi^\theta} \frac{1}{(n-1)^2}. \end{aligned}$$

□

PROPOSITION 7.2 *Let T_n, N_k, k, θ and d as in the previous proposition. Then, for all $\xi > 0$, there is $\varphi \in (0,1)$, such that*

$$\mathbf{P}\left\{M_{n,1} > (n-1)(\alpha+\xi), N_k \leq (n-1)\xi/2\right\} \leq \varphi^{(n-1)^d},$$

for all n large enough.

PROOF. Denote by $v_1, \ldots, v_{b^k}$ all nodes in T_n at depth k. Define B^i_{n-k} to be the minimum of all $V_{v_i,w}$ such that w is a leaf descendant of v_i. Then, by independence of the B^i_{n-k}'s,

$$\begin{aligned} &\mathbf{P}\left\{M_{n,1} > (n-1)(\alpha+\xi),\ N_k \leq (n-1)\xi/2\right\} \\ &\leq \mathbf{P}\left\{B^1_{n-k} + (n-1)\xi/2 > (n-1)(\alpha+\xi), \ldots, B^{b^k}_{n-k} + (n-1)\xi/2 > (n-1)(\alpha+\xi)\right\} \\ &= \left(\mathbf{P}\left\{B_{n-k} > (n-1)(\alpha+\xi/2)\right\}\right)^{b^k} \\ &\leq \left(\mathbf{P}\left\{B_{n-k} > (n-k)(\alpha+\xi/2)\right\}\right)^{b^k}, \end{aligned}$$

where B_{n-k} is the random variable defined in theorem 6.2. Substituting k by $\lfloor d \log_b(n-1)\rfloor$ and using theorem 6.2 we get that

$$\mathbf{P}\left\{B_{n-\lfloor d\log_b(n-1)\rfloor} > (n - \lfloor d\log_b(n-1)\rfloor)(\alpha+\xi/2)\right\} \longrightarrow 0, \quad \text{as } n \to \infty.$$

Thus we can find $\varphi \in (0,1)$ such that

$$\mathbf{P}\left\{B_{n-\lfloor d\log_b(n-1)\rfloor} > (n-\lfloor d\log_b(n-1)\rfloor)(\alpha+\xi/2)\right\} \le \varphi,$$

for all n large enough. Hence,

$$\begin{aligned}
&\mathbf{P}\left\{M_{n,1} > (n-1)(\alpha+\xi),\, N_k \le (n-1)\xi/2\right\} \\
&\le \left(\mathbf{P}\left\{B_{n-\lfloor d\log_b(n-1)\rfloor} > (n-\lfloor d\log_b(n-1)\rfloor)(\alpha+\xi/2)\right\}\right)^{(n-1)^d} \\
&\le \varphi^{(n-1)^d},
\end{aligned}$$

for all n large enough. □

The following corollary is immediate from the two previous propositions.

COROLLARY 7.1 *Assume* $\mathbf{E}\left\{X^\theta\right\} < \infty$ *for some* $\theta > 2$. *Pick* $d \in (0, \theta-2)$. *For all* $\xi > 0$, *there are* $\zeta > 0$, $\varphi \in (0,1)$, *and* n_0 *such that*

$$\mathbf{P}\left\{M_{n,1} > (n-1)(\alpha+\xi)\right\} \le \frac{\zeta}{(n-1)^2} + \varphi^{(n-1)^d}, \text{ for all } n \ge n_0.$$

7.4 Proof of main theorem

We first prove one half of part A of theorem 7.1. The second half of part A follows from part B, and will be proved in theorem 7.3 below.

THEOREM 7.2. *Let* X *be a regular random variable with* $\mathbf{E}\left\{X^\theta\right\} < \infty$ *for some* $\theta > 2$. *Then for every* $\varepsilon > 0$

$$\lim_{n\to\infty} \mathbf{P}\left\{N^{1/n} > \beta+\varepsilon\right\} = 0.$$

PROOF. Let T_n be the random b-ary tree as defined in the previous section. It is clear that the number descendants of v_{kl} visited by the algorithm is smaller than or equal to $bZ_{kl}+1$, because if the value of a node is less than

the minimal leaf value seen thus far, all its b children will be visited. It follows that

$$N \leq b\left(\sum_{k=1}^{n}\sum_{l=1}^{b-1} Z_{kl} + n\right) + 1.$$

Thus,

$$\begin{aligned}\mathbf{P}\left\{N^{1/n} > \beta+\varepsilon\right\} &= \mathbf{P}\left\{N > (\beta+\varepsilon)^n\right\} \\ &\leq \mathbf{P}\left\{b\sum_{k=1}^{n}\sum_{l=1}^{b-1} Z_{kl} + bn + 1 > (\beta+\varepsilon)^n\right\} \\ &\leq \sum_{k=1}^{n}\sum_{l=1}^{b-1}\mathbf{P}\left\{b\,Z_{kl} > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\},\end{aligned}$$

if $(\beta+\varepsilon)^n/((b-1)n+1) \geq bn+1$, which is true for all n large enough. Now, notice that $Z_{kl} \leq 1 + b + \cdots + b^{n-k} \leq b^{n-k+1}$, so that for $k \geq \lceil(1-\delta)n\rceil$,

$$bZ_{kl} \leq b^{n-k+1} \leq b^{\delta n+1} \leq \frac{(\beta+\varepsilon)^n}{(b-1)n+1},$$

where δ is taken

$$0 < \delta \leq \log_b(\beta+\varepsilon) - \frac{\log_b\left((b-1)n+1\right)+1}{n}.$$

Observe that the previous can be done for all n large enough. The previous observation implies that

$$\sum_{k=1}^{n}\sum_{l=1}^{b-1}\mathbf{P}\left\{b\,Z_{kl} > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} \leq \sum_{k=1}^{\lfloor(1-\delta)n\rfloor}\sum_{l=1}^{b-1}\mathbf{P}\left\{b\,Z_{kl} > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} \qquad (*)$$

for all n large enough. Next, for all $\xi > 0$,

$$\begin{aligned}&\mathbf{P}\left\{b\,Z_{kl} > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} \\ &\qquad \leq \mathbf{P}\left\{M_{n,k} > (n-k)(\alpha+\xi)\right\} + \mathbf{P}\left\{b\,Z_{kl}\,(\alpha+\xi) > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\},\end{aligned}$$

where the random variable $Z_{kl}\,(\alpha+\xi)$ represents the number of descendants w of v_{kl}, such that $V_{u_{k-1},w}$ is not larger than $(\alpha+\xi)(n-k)$. Corollary 7.1 implies that there are $\zeta > 0$, $\varphi \in (0,1)$, and n_0, such that,

$$\mathbf{P}\left\{M_{n,k} > (n-k)(\alpha+\xi)\right\} \leq \zeta/(n-k)^2 + \varphi^{(n-k)^d},$$

for $n \geq n_0$, uniformly over $1 \leq k \leq (1-\delta)n$. Thus, for $n \geq n_0$,

$$\begin{aligned}
\sum_{k=1}^{\lfloor (1-\delta)n \rfloor} \sum_{l=1}^{b-1} \mathbf{P}\left\{M_{n,k} > (n-k)\,(\alpha+\xi)\right\} &\leq \sum_{k=n-\lfloor (1-\delta)n \rfloor}^{n-1} b\left(\frac{\zeta}{k^2} + \varphi^{k^d}\right) \\
&\leq \frac{bn\zeta}{(n-\lfloor (1-\delta)n \rfloor)^2} + b\varphi^{(n-\lfloor (1-\delta)n \rfloor)^d} \lfloor (1-\delta)n \rfloor \\
&\leq \frac{b\zeta}{\delta^2 n} + b(1-\delta)\varphi^{(\delta n)^d} n \\
&\longrightarrow 0 \qquad \text{as } n \to \infty.
\end{aligned}$$

Now, for the other part of the sum in $(*)$, note that

$$\begin{aligned}
\mathbf{P}\left\{b\, Z_{kl}\,(\alpha+\xi) > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} &\leq \frac{\mathbf{E}\left\{b\, Z_{kl}\,(\alpha+\xi)\right\}}{(\beta+\varepsilon)^n}\,((b-1)n+1) \\
&= \frac{\mathbf{E}\left\{b \sum_{j=1}^{n-k} Z_{kl}^{j}\,(\alpha+\xi)\right\}((b-1)n+1)}{(\beta+\varepsilon)^n},
\end{aligned}$$

where $Z_{kl}^{j}\,(\alpha+\xi)$ is the number of descendants w of v_{kl} such that $V_{u_{k-1},w}$ is smaller than or equal to $(\alpha+\xi)(n-k)$, and the distance from w to v_{kl} is j. Now, for $\theta > 0$,

$$\begin{aligned}
&\mathbf{E}\left\{Z_{kl}^{j}\,(\alpha+\xi)\right\} \leq b^j\, \mathbf{P}\left\{X_1 + \cdots X_j + X_{j+1} \leq (n-k)(\alpha+\xi)\right\} \\
&\leq b^j\, \mathbf{E}\left\{e^{\theta((n-k)(\alpha+\xi) - X_1 - \cdots - X_{j+1})}\right\} \text{ (by Markov's inequality)} \\
&= b^j e^{\theta(n-k)(\alpha+\xi)} \prod_{i=1}^{j+1} \mathbf{E}\left\{e^{-X_i\theta}\right\} \text{ (by the independence of the } X_i\text{'s)} \\
&= b^j e^{\theta(n-k)(\alpha+\xi)} \left(\mathbf{E}\left\{e^{-X\theta}\right\}\right)^{j+1} \text{ (because the } X_i\text{'s are identically distributed)} \\
&\leq \left(b\,\mathbf{E}\left\{e^{\theta\left(\frac{(n-k)(\alpha+\xi)}{j+1} - X\right)}\right\}\right)^{j+1} \\
&= \left(\mu\left(\frac{(n-k)(\alpha+\xi)}{j+1}\right)\right)^{j+1},
\end{aligned}$$

where for the last equality to hold we took $\theta = \theta^*$ such that

$$b\,\mathbf{E}\left\{e^{\theta^*\left(\frac{(n-k)(\alpha+\xi)}{j+1} - X\right)}\right\} = \mu\left(\frac{(n-k)(\alpha+\xi)}{j+1}\right).$$

Thus, since $\mu(a)$ is an increasing function of $a \geq 0$,

$$\begin{aligned}
\mathbf{P}\left\{b\, Z_{kl}(\alpha+\xi) > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} &\leq \frac{b((b-1)n+1)}{(\beta+\varepsilon)^n} \sum_{j=1}^{n-k} \mathbf{E}\left\{Z_{kl}^j(\alpha+\xi)\right\} \\
&\leq \frac{b((b-1)n+1)}{(\beta+\varepsilon)^n} \sum_{j=1}^{n-k} \left(\mu\left(\frac{(n-k)(\alpha+\xi)}{j+1}\right)\right)^{j+1} \\
&= \frac{b((b-1)n+1)}{(\beta+\varepsilon)^n} \sum_{j=1}^{n-k} \left(\mu^{\frac{j+1}{n+1}}\left(\frac{(n+1)(\alpha+\xi)}{j+1}\right)\right)^{n+1} \\
&\leq \frac{b((b-1)n+1)}{(\beta+\varepsilon)^n} \sum_{j=1}^{n-k} \left(\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi}{x}\right)\right)^{n+1} \\
&\leq \frac{nb((b-1)n+1)}{(\beta+\varepsilon)^n} \left(\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi}{x}\right)\right)^{n+1}.
\end{aligned}$$

We now sum all the terms and get

$$\begin{aligned}
&\sum_{k=1}^{\lfloor(1-\delta)n\rfloor} \sum_{l=1}^{b-1} \mathbf{P}\left\{b\, Z_{kl}(\alpha+\xi) > \frac{(\beta+\varepsilon)^n}{(b-1)n+1}\right\} \\
&\qquad \leq \frac{n^2 b(b-1)((b-1)n+1)}{(\beta+\varepsilon)^n} \left(\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi}{x}\right)\right)^{n+1} \\
&\qquad \leq \frac{b^3 n^3}{(\beta+\varepsilon)^n} \left[\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi}{x}\right)\right]^{n+1}.
\end{aligned}$$

Notice that if ξ were equal to 0, then the quantity in the square brackets would be β^{n+1}. Lemma 6.1 implies that for $\varepsilon > 0$, there exists $\xi^* > 0$ such that

$$\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi^*}{x}\right) \leq \sup_{x\in(0,1)} \mu^x\left(\frac{\alpha}{x}\right) + \frac{\varepsilon}{2} = \beta + \frac{\varepsilon}{2}.$$

Therefore,

$$\begin{aligned}
&\frac{b^3 n^3}{(\beta+\varepsilon)^n} \left(\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi^*}{x}\right)\right)^{n+1} \leq \\
&\qquad b^3 n^3 \left(\frac{\sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi^*}{x}\right)}{\beta+\varepsilon}\right)^n \sup_{x\in(0,1)} \mu^x\left(\frac{\alpha+\xi^*}{x}\right) \\
&\qquad \leq b^3 n^3 \left(\frac{\beta+\varepsilon/2}{\beta+\varepsilon}\right)^n (\beta+\varepsilon/2) \\
&\qquad \underset{n\to\infty}{\longrightarrow} 0.
\end{aligned}$$

□

To finish the proof of theorem 7.1, we show the universal lower bound of part B. Note that any algorithm must visit all nodes v in T_n with value strictly less than B_n, the minimal leaf value in T_n. Thus, it suffices to prove the following.

THEOREM 7.3. *Let $X \geq 0$ be a regular random variable. Let N' be the number of nodes v in T_n with value $V_v < B_n$. Then, for every $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbf{P}\left\{N'^{1/n} < \beta - \epsilon\right\} = 0.$$

PROOF. We use the notation from theorem 6.1, and note that

$$N' \geq \sum_{j=0}^{n} Z^{(j)}(B_n - 1).$$

Define α, β and γ as in the previous chapter, and set $k = \lfloor \gamma n \rfloor$. Thus, for any $\xi \in (0, \alpha)$,

$$N' \geq Z^{(k)}\left((\alpha - \xi)n\right) I_{[B_n - 1 > (\alpha-\xi)n]}.$$

By theorem 6.2, with probability one, $B_n - 1 > (\alpha - \xi)n$ for all n large enough. Thus, we are done if we can show that given $\epsilon > 0$, we can find $\xi > 0$ such that

$$\lim_{n\to\infty} \mathbf{P}\left\{\left(Z^{(k)}\left((\alpha - \xi)n\right)\right)^{1/n} < \beta - \epsilon\right\} = 0.$$

To this effect, observe that

$$\begin{aligned}
\left(Z^{(k)}\left((\alpha - \xi)n\right)\right)^{1/n} &= \left[\left(Z^{(k)}\left(\frac{(\alpha-\xi)n}{k}k\right)\right)^{k/n}\right]^{1/k} \\
&\geq \left[\left(Z^{(k)}\left(\frac{\alpha-\xi}{\gamma}k\right)\right)^{\gamma-\frac{1}{n}}\right]^{1/k} \\
&\to \left[\mu\left(\frac{\alpha-\xi}{\gamma}\right)\right]^{\gamma}
\end{aligned}$$

by theorem 6.1. By the continuity of μ, the lower bound tends to β as $\xi \downarrow 0$, since $\beta = \mu^{\gamma}(\alpha/\gamma)$. □

7.5 Some examples.

b	ρ	b	ρ
2	.334648	3	.257101
4	.220361	5	.198027
6	.182672	7	.171302
8	.162452	9	.155311
10	.149393	11	.144383
12	.140071	13	.136306
14	.132983	15	.130019
16	.127354	17	.124941
18	.122741	19	.120725
20	.118868	21	.117149

TABLE 7.1. Values of b vs. ρ for the exponential distribution.

In this section we will present some examples of μ-functions and β-values for some well-known distributions.

EXAMPLE. Exponential Distribution. If X is exponentially distributed, $m(\theta) = 1/(\theta+1)$, and

$$\mathbf{E}\left\{e^{\theta(a-X)}\right\} = \frac{e^{\theta a}}{\theta+1},$$

for $\theta \geq 0$. Also,

$$\mu(a) = b \inf_{\theta \geq 0}\left\{\frac{e^{\theta a}}{\theta+1}\right\} = \begin{cases} bae^{1-a}, & a \leq 1; \\ b \quad , & \text{otherwise,} \end{cases}$$

because $\log\left(\frac{be^{\theta a}}{\theta}\right)$ is minimum for $\theta = \frac{1}{a} - 1$. The value α is defined to be the unique solution of

$$\alpha b e^{1-\alpha} = 1. \tag{2}$$

Observe that as $b \to \infty$, $\alpha \sim 1/2b$. Note that

$$\begin{aligned} \sup_{x\in(0,1)} \mu^x\left(\frac{\alpha}{x}\right) &= \sup_{x\in(\alpha,1)}\left\{\frac{\alpha}{x} b e^{1-\frac{\alpha}{x}}\right\}^x \vee \sup_{x\in(0,\alpha)} b^x \\ &= \sup_{x\in(\alpha,1)}\left\{\frac{e^{\alpha x}}{x^x} e^{-\alpha}\right\} \vee b^\alpha, \end{aligned}$$

b	ρ	b	ρ
2	.590941	3	.455860
4	.372014	5	.311999
6	.265677	7	.228184
8	.196821	9	.169940
10	.146465	11	.125653
12	.106973	13	.090028
14	.074516	15	.061973
16	.046876	17	.034384
18	.022566	19	.011245

TABLE 7.2. Values of b vs. ρ for the Bernoulli (.05) distribution.

because $\alpha b = e^{\alpha-1}$. The value of x which maximizes the first quantity is $e^{\alpha-1}$ (which is in $(\alpha, 1)$). By manipulating (2) we get

$$\begin{aligned}\beta &= \left\{ e^{-\alpha} \frac{e^{\alpha e^{\alpha-1}}}{e^{\alpha e^{\alpha-1}} e^{-e^{\alpha-1}}} \right\} \vee b^{\alpha} \\ &= \max\left\{ e^{e^{\alpha-1}-\alpha}, b^{\alpha} \right\} \\ &= e^{\alpha(b-1)},\end{aligned}$$

because $e^{\alpha-1} = \alpha b$. Note that $e^{\alpha(b-1)} < e$ for any value of b, and $\beta \to \sqrt{e}$ as $b \to \infty$. Also, $\rho = \log_b \beta \sim 1/(2 \log b)$, as $b \to \infty$.

EXAMPLE. Bernoulli distribution with parameter $p \in (0, 1)$. Let X be a Bernoulli(p) distributed random variable, then

$$\mathbf{E}\left\{e^{-\theta X}\right\} = p + (1-p)e^{-\theta},$$

and

$$\begin{aligned}\mu(a) &= b \inf_{\theta \geq 0} \mathbf{E}\left\{e^{\theta(a-X)}\right\} \\ &= b \inf_{\theta \geq 0} e^{\theta a}(p + (1-p)e^{\theta}) \\ &= \begin{cases} b\left(\frac{p}{1-a}\right)^{1-a}\left(\frac{1-p}{a}\right)^{a}, & 0 < a \leq 1-p, \\ b, & a \geq 1-p. \end{cases}\end{aligned}$$

b	ρ	b	ρ
2	.522782	3	.445452
4	.405182	5	.379173
6	.360477	7	.346137
8	.334648	9	.325150
10	.317110	11	.310176
12	.304107	13	.298731
14	.293919	15	.289575
16	.285625	17	.282010
18	.278682	19	.275604
20	.272745	21	.270078

TABLE 7.3. Values of b vs. ρ for the gamma (3) distribution.

So as to define α, we must assume that $bp < 1$. Thus α is defined as the solution of the equation

$$b\left(\frac{p}{1-\alpha}\right)^{1-\alpha}\left(\frac{1-p}{\alpha}\right)^{\alpha} = 1.$$

Note that as $p = \mathbf{P}\{X = 0\}$ and taking $b = 2$ we recuperate one of the results of Karp and Pearl (1983).

EXAMPLE. Gamma distribution with parameter r. We first compute

$$\begin{aligned}
\mu(a) &= b \inf_{\theta \geq 0} \int_0^{\infty} \frac{x^{r-1}e^{-x}e^{\theta(a-x)}}{\Gamma(r)} dx \\
&= b \inf_{\theta \geq 0} \frac{e^{\theta a}}{(1+\theta)^r} \\
&= b\frac{e^{a\left(\frac{r}{a}-1\right)}}{\left(\frac{r}{a}\right)^r} \vee b \\
&= \begin{cases} be^{r-a}\left(\frac{a}{r}\right)^r, & \text{if } 0 \leq a \leq r; \\ b \qquad\qquad , & a \geq r. \end{cases}
\end{aligned}$$

because $\theta a - r\log(1+\theta)$ is minimal when $a = \frac{r}{1+\theta}$. Thus α is defined as the solution to the equation:

$$be^{r-\alpha}\left(\frac{\alpha}{r}\right)^r = 1.$$

Finally,

$$\beta = \sup_{x \in (0,1)} \mu^x\left(\frac{\alpha}{x}\right) = \sup_{x \in (0,1)} b^x e^{rx-\alpha}\left(\frac{\alpha}{rx}\right)^{rx}.$$

Note that $\mu^x\,(\alpha/x)$ is maximal at $x = b^{1/r}/r$ and therefore

$$\beta = \frac{b^{b^{1/r}\alpha/r} e^{\alpha(b^{1/r}-1)}}{b^{b^{1/r}\alpha/r}} = e^{\alpha(b^{1/r}-1)}.$$

Conclusions

In the first part of the thesis we studied the expected time complexity of range search when the data structure used for storing the data is the k-d tree. By studying the geometry of the rectangle partition generated by the k-d tree, we were able to find a tight expression for the expected time complexity of range search that reflected the geometry of the query region. This result showed the way to improve on the expected complexity of range search by defining a new data structure that we called squarish k-d tree.

In chapter 3 and 4 we saw that k-d trees are not optimal even in an average sense for solving range search. The elongated rectangles in the partition generated using k-d trees explain its poor performance. We showed that squarish k-d trees behave optimally in an expected sense. For instance, that the expected time complexity of partial match in 2-d trees, when specifying one attribute, is $\Theta\left(n^{\frac{\sqrt{17}-3}{2}}\right) = \Theta\left(n^{0.561552\ldots}\right)$, whereas for 2-d squarish trees it is $\Theta(\sqrt{n})$.

In chapter 5 we analyzed two natural algorithms for solving the nearest neighbor problem when using k-d trees. We conjectured that for k-d trees the expected time complexity of nearest neighbor queries is $\Omega(n^{\rho_k})$. We also conjecture that the expected time complexity of nearest neighbor queries when using algorithm **A** and squarish k-d trees is indeed $O(\log n)$. This requires further research.

Another very interesting problem that deserves further study is the expected worst-case complexity of range search. That is, the data points are still random, but now the position of the query can be chosen arbitrarily. We believe that the expected worst case complexity over all partial match queries with worst case location of the free corrdinates, is also bounded from above

by the bound given in theorem 3.2 and 3.4 for the k-d tree and squarish k-d tree respectively.

In the second part of the thesis we studied the time complexity of branch-and-bound search for random trees. Theorem 7.1 shows that up to first-order asymptotics, ordinary depth-first search is as good as any search strategy for all regular random variables. This is remarkable, as depth-first search can be implemented using only $O(n)$ storage.

Bibliography

ABRAMOWITZ, M. AND STEGUN, I. A. (1970), *Handbook of Mathematical Functions*, Dover Publications, New York.

AGARWAL, P. K. (1997), *Handbook of Discrete and Computational Geometry*, CRC Press, chapter Range searching.

ASMUSSEN, S. AND HERING, H. (1983), *Branching Processes*, Birkhäuser, Boston.

ATHREYA, K. B. AND NEY, P. E. (1972), *Branching Process*, Springer-Verlag, Berlin.

BENTLEY, J. L. (1975), Multidimensional binary search trees used for associative searching, *Communications of the ACM* **18**, 509–517.

BENTLEY, J. L. (1979), Multidimensional binary search trees in database applications, *IEEE Transactions on Software Engineering* **SE-5**, 333–340.

BENTLEY, J. L., AND FINKEL, J. H. (1979), Data structures for range searching, *ACM Computing Surveys* **11**, 397–409.

BENTLEY, J. L. AND STANAT, D. F. (1975), Analysis of range searches in quad trees, *Information Processing Letters* **3**, 170–173.

BENTLEY, J. L. (1990), K-d trees for semidynamic point sets, 6^{th} *Annual Symposium on Computational Geometry*, 187–197.

BENTLEY, J. L., WEIDE, W., AND YAO, A. C. (1980), Optimal expected time algorithms for closest point problems, *ACM Transactions on Mathematical Software*, **6**, 563–580.

BIGGINS, J. D. (1977), Chernoffs theorem in the branching random walk, *Journal of Applied Probability* **14**, 630–636.

BROWN, C. A. AND PURDOM, P. W. (1981), An average time analysis of backtracking, *SIAM Journal of Computing* **10**, 583–593.

CHANZY, P., DEVROYE, L. AND ZAMORA-CURA, C. (1999), Analysis of range search for random k-d trees. Submitted for publication.

CUNTO, W., LAU, G. AND FLAJOLET, P. (1989), Analysis of kdt-trees: kd-trees improved by local reorganizations, in F. Dehne, J. R. Sack and N. Santoro, ed, *Workshop on Data Structures and Algorithms (WADS89)*, vol. 382 LNCS, Springer Verlag, 24–38.

DEVROYE, L. (1986), A note on the height of binary search trees, *Journal of the ACM* **33**, 489–498.

DEVROYE, L. (1987), Branching processes in the analysis of the heights of trees, *Acta Informatica* **24**, 277–298.

DEVROYE, L., JABBOUR, J. AND ZAMORA-CURA, C. (1999), Squarish k-d trees. Submitted to the *SIAM Journal of Computing*.

DEVROYE, L. AND LAFOREST, L. (1990), An analysis of random d-dimensional quadtrees, *SIAM Journal on Computing* **19**, 821–832.

DEVROYE, L. AND ZAMORA-CURA, C. (1999), On the complexity of branch-and-bound search for random trees, *Random Structures and Algorithms*, **14**, 309–327.

DOBKIN, D AND LIPTON, R. J. (1976), Multidimensional searching problems, *SIAM Journal of Computing* **5**, 181–186.

DUCH, A., ESTIVILL-CASTRO, V. AND MARTINEZ, C. (1998), Randomized k-dimensional binary search trees, Technical Report Universitat Polytècnica de Catalunya, LSI-98-48.ps.

FINKEL, R. A. AND BENTLEY, J. L. (1974), Quad trees: a data structure for retrieval on composite keys, *Acta Informatica* pp. 1–9.

FLAJOLET, P., GONNET, G. AND PUECH, C. (1991), The analysis of multidimensional searching in quad-trees, *in Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, pp. 100–109.

FLAJOLET, P., GONNET, G. AND PUECH, C. (1992), Analytic variations on quadtrees, *Algorithmica* **10**, 473–500.

FLAJOLET, P., GONNET, G., PUECH, C. AND ROBSON, J. M. (1993), Analytic variations on quadtrees, *Algorithmica*, **10**, 473–500.

FLAJOLET, P. LAFFORGUE, T. (1994), Search costs in quadtrees and singularity perturbation analysis, *Discrete and Computational Geometry* **12**, 151–175.

FLAJOLET, P. AND PUECH, P. (1986), Partial match retrieval of multidimensional data, *Journal of the ACM* **33**, 371–407.

FRIEDMAN, J. H., BENTLEY, J. L. AND FINKEL, R. A. (1977), Al algorithm for finding best matches in logarithmic expected time, *ACM Tansactions on Mathematical Software*, **3**, 209–226.

FUKUNAGA, K. AND NARENDRA, P. M. (1975), A branch-and-bound- algorithms for computing the k-nearest neighbors, *IEEE Transactions of Computing*, **C-24**, 750–753.

GARDY, D., FLAJOLET, P. AND PUECH, C. (1989), Average cost of orthogonal range queries in multiattribute trees, *Information Systems* **14**, 341–350.

GONNET, G. H. AND BAEZA-YATES, R. (1991), *Handbook of Algorithms and Data Structures*, Addison-Wesley, Workingham.

HAMMERSLEY, J. M. (1974), Postulates for subadditive processes, *Annals of Probability* **2**, 652–680.

HARRIS, T. E. (1963), *The Theory of Branching Processes*, Springer-Verlag.

JOHN, F. (1948), *Studies and Essays Presented to R. Courant*, Interscience, New York, chapter Extremum problems with inequalities as subsidiary conditions, pp. 187–204.

KARP, R. M. AND PEARL, J. (1983), Searching for an optimal path in a tree with random costs, *Artificial Intelligence* **21**, 99–117.

KINGMAN, J. F. C. (1975), The first-birth problem for an age-dependant branching process, *Annals of Probability* **3**(5), 341–345.

KNUTH, D. E. (1997), *The Art of Computer Programming*, Vol. 3. 2nd ed., Addison-Wesley, Reading, MA.

KUMAR, V. (1992). *Encyclopedia of Artificial Intelligence*, 2nd ed., Wiley-Interscience, chapter Search, branch and bound, pp. 1468–1472.

LEE, D. T. AND WONG, C. K. (1977), Worst-case analysis for region and partial region searches in multidimensional binary search trees and quad trees, *Acta Informatica* **9**, 23–29.

MAHMOUD, H. M. (1992), *Evolution of Random Search Trees*, John Wiley, New York

MARTINEZ, C., PANHOLZER, A. AND PRODINGER, H. (1998), Partial match queries in relaxed multidimensional search trees, Technical Report Universitat Polytècnica de Catalunya, LSI-98-53.ps.

MATOUSEK, J. (1994), Geometric range searching, *ACM Computing Surveys* **26**, 421–461.

MCDIARMID, C. J. H. (1990), *Disorder in Physical Systems*, Oxford Science Publications, chapter Probabilistic analysis of tree search, pp. 249–260.

McDiarmid, C. J. H. and Provan, G. M. A. (1991), An expected-cost analysis of backtracking and non-backtracking algorithms, *in IJCAI-91: Proceedings of the Twelfth International Conference on Artificial Intelligence*, Morgan Kaufmann Publishing, San Mateo, CA, pp. 172–177.

Mitrinovic, D. S. (1970), *Analytic Inequalities*, Springer-Verlag, New York.

Neininger, R. (1999), Asymptotic distributions for partial match queries in k-d trees, preprint, Universität Freiburg.

Neininger, R. and Rüschendorf, L., (1999), Limit laws for partial match queries in quadtrees, preprint, Universität Freiburg.

Pearl, J. (1984), *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading, Mass.

Pittel, B. (1984), On growing random binary trees, *Journal of Mathematical Analysis and Applications* **103**, 461–480.

Purdom, P. W. (1983), Search rearrangement backtracking and polynomial average time, *Artificial Intelligence* **21**, 117–133.

Rabin, M (1976), Probabilistic algorithms, in *Algorithms and Complexity*, ed. J. Traub. Academic Press, New york, N.Y., 21–39.

Reingold, E. M., Nievergelt, J. and Deo, N. (1977), *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, N.J.

Robson, J. M. (1979), The height of binary search trees, *The Australian Computer Journal* **11**, 151–153.

Samet, H. (1990*a*), *Applications of Spatial Data Structures*, Addison-Wesley, Reading, MA.

Samet, H. (1990*b*), *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA.

Shamos, M. I. and Hoey, D. (1975), Closest-point problems, *IEEE Symp. on Foundations of Computer Science*, 151–162.

SMITH, D. R. (1984), Random trees and the analysis of branch and bound procedures, *Journal of the ACM* pp. 163–188.

SPROUL, R. (1991), Refinements to nearest neighbor searching in k-dimensional trees, *Algorithmica*, (6), 579–589.

STONE, H. S. AND SIPALA, P. (1986), The average complexity of depth-first search with backtracking and cutoff, *IBM Journal of Research and Development* **30**, 242–258.

VITTER, J. AND FLAJOLET, P. (1990), *Handbook of Theoretical Computer Science*, Vol. A: Algorithms and Complexity, MIT Press, chapter Average-case analysis of algorithms and data structures.

VAN KREVELD, M. AND OVERMARS, M. (1991), Divided k-d trees, *Algorithmica*, (6), 840–858.

WAH, B. W. AND YU, C. F. (1985), Stochastic modeling of branch-and-bound algorithms with best-first search, *IEEE Transactions of Software Engineering* **SE-11**, 922–934.

WHITTAKER, E. T. AND WATSON, G. N. (1927), *A Course of Modern Analysis*, Cambridge University Press, Cambridge, U.K.

YAO, F. F. (1990), *Handbook of Theoretical Computer Science*, Vol. A: Algorithms and Complexity, MIT Press, Amsterdam, chapter Computational geometry, pp. 343–389.

YUVAL, G. (1976), Finding nearest neighbors, *Information Processing Letters*, **5**, 63–65.

ZHANG, W. AND KORF, R. E. (1992), An average-case analysis of branch-and-bound with applications, *in Proceedings of the 10th National Conference on AI—AAAI-92*, San Jose, CA, pp. 1–6.