Space-time tradeoffs for orthogonal range queries

(Extended Abstract)

Pravin M. Vaidya

Department of Computer Science

University of Illinois at Urbana-Champaign

Urbana, IL 61801

Abstract

We investigate the question of (storage) space - (retrieval) time tradeoff for orthogonal range queries on a static database. Lower bounds on the product of retrieval time and storage space are obtained in the arithmetic and tree models.

1. Introduction

Consider a data base that contains a collection of records, each with a key and a number of data fields. Given a range query, which is specified by a set of constraints on the keys, the data base system is expected to return the set of records, or a function of the set of records whose keys satisfy all the constraints. If the data base is static the collection of records may be preprocessed to achieve a balance between the storage utilised and the time required to answer a query. There is an extensive literature [1, 2, 3, 7, 8, 9, 10] on algorithms for range query, and the space and time requirements have traditionally been used as performance measures for such algorithms. In this paper, we investigate the question of (storage) space - (retrieval) time tradeoff for orthogonal range queries on a static database.

Let G be a commutative semigroup with an addition operation $+$. Let d be a fixed positive integer. Let $N = \{1,2,....,n\}$ and let N^d denote the set of all d-tuples of positive integers less than or equal to n. A record $(k,f(k))$ is a pair of key $k \in N^d$ and datum $f(k) \in G$. The data base consists of n such records. Let $k = (k_1, k_2, ..., k_d)$. An orthogonal range query is specified by a $2d$-tuple $q = (x_{11}, x_{12}, x_{21}, x_{22}, ..., x_{d1}, x_{d2})$ of positive integers satisfying $x_{i1} < x_{i2}$, $1 \leq i \leq d$, or alternately, the query region is a parallelepiped (box) b, defined by the product $[x_{11}, x_{12}) \times [x_{21}, x_{22}) \times .... \times [x_{d1}, x_{d2})$ of d-semiclosed intervals with positive integer endpoints. We consider two types of response to such a query, one where the output is the semigroup sum of the data $f(k)$ whose key k are located in the query parallelepiped (box) b, and the other where the output is the list of records whose keys lie in the query parallelepiped b. We use $Q(b)$ to denote the input tuple q corresponding to query region b, and K to denote the set of keys in the database.

A space-time tradeoff seeks to answer questions like what is the minimum amount of storage needed to ensure a certain query time. In the orthogonal range query problem the set of query regions is fixed. The tradeoff between space and time is dependent on the model of data structure and the set of records in the data base. We fix the data structure model and then try to obtain a set of records that makes this tradeoff as worse as possible. Thus the lower bounds on space-time products are to be intrepreted as worst case bounds, i.e. there exists a set of n records whose space-time product has the said bounds.

We study two models. In Model A, we work in the general framework defined by Fredman [4, 5, 6], and consider only data structures and manipulation algorithms which are independent of the choice of the semigroup G. So the set K of keys in the database together with the set of query regions completely specifies the problem. The model is an arithmetic model with unit cost for each arithmetic operation but no cost for memory retrieval. We only consider that type of response where the output is the semigroup sum of the data values whose keys lie in the query region. In this model, we show that for orthogonal range query on a static database with n records, there is a space-time tradeoff $TS \geq \Omega(n(\log_T n)^{d-\theta})$ where $\theta = 1$ for $d = 2,3$, and $\theta = 2$ for $d > 3$. Space-time tradeoffs for circular range query and interval query in this model are studied by Yao in [11]. The complexity of dynamic range queries in this model is discussed by Fredman in [4, 5, 6].

In Model B (tree model), we study a broad class of rooted tree data structures. In this model, the data structures are no longer independent of the choice of the semigroup G, and any correlation between the distributions of the given keys and the corresponding data values can be utilised to build more efficient data structures. We investigate two instances of orthogonal range query, the *counting problem* and the *maximum problem*, where the output is a single element in the semigroup G. In the counting problem, the response to a query is the number of keys located in the query parallelepiped. In the maximum problem, there is a linear ordering on the data values and the response to a query is the maximum of the data values whose keys lie in

*This work was supported by a fellowship from the Shell Foundation

© 1985 ACM 0-89791-151-2/85/005/0169 $00.75

the query parallelepiped. For the counting problem, for a static data base with n records, we show that there is a space-time tradeoff $T\,S \geq \Omega(n(\log_T n)^{d-\theta})$ where $\theta = 1$ for $d = 2,3$ and $\theta = 2$ for $d > 3$. For the maximum problem, we show that $(\log T + \log\log n)^{d-1}\, S \geq \Omega(n(\log n)^{d-\theta})$ where $\theta = 1$ for $d = 2,3$, and $\theta = 2$ for $d > 3$.

In Model B, we also investigate the *reporting problem* where the response to a query is the list of all the records whose keys are located in the query parallelepiped. For a set of records and a corresponding tree data structure, let $L(b)$ be the number of records whose keys lie in the query region b, let $T(b)$ be the time required to answer the query corresponding to b, and let

$$T' = \max_{b \text{ such that } 1 \leq L(b) \leq \log_2 n} \frac{T(b)}{L(b)}.$$

For the reporting problem we show that there is a space-time tradeoff $(\log T' + \log\log n)^{d-1}\, T'\, S \geq \Omega(n(\log n)^{d-\theta})$, where $\theta = 1$ for $d = 2,3$, and $\theta = 2$ for $d > 3$.

2. An Overview

2.1. Model A - Arithmetic Model

In this model [], a data structure utilises an infinite array Z of variables $z_0, z_1, z_2, \ldots$, which stores elements from the commutative semigroup G. Given any input query, the query answering algorithm chooses a collection of at most m variables in the array and returns their semigroup sum as the response to the query. Since only arithmetic operations are charged, the smallest possible m for which the query answering algorithm works correctly is the query time T. The data structure is assumed to be independent of the specific semigroup G and so the mapping between elements in G and variables in Z is determined solely by the set K of n keys in the data base. With each variable z_i in Z is associated a subset h_i of K and the data value $\sum_{k \in h_i} f(k)$ is stored in z_i, where $f(k)$ is the data value associated with k. Let $H \subseteq 2^K$ such that every set in H is associated with some variable in Z and every variable in Z is associated with some set in H. Let R be the set of all possible query regions, and $P(H,T)$ be the property that for each $b \in R$, $b \cap K$ is expressible as the disjoint union of at most T sets in H. The query answering algorithm works correctly iff $P(H,T)$ is satisfied. For a set K of keys, the storage space S is defined by

$$S = \min_{H \text{ satisfying } P(H,T)} |H|.$$

The following Theorem summarizes the results in this model.

Theorem 1. In Model A, for orthogonal range query on a static database with n records, there is a space-time tradeoff $T\,S \geq \Omega(n(\log_T n)^{d-\theta})$ where $\theta = 1$ for $d = 2,3$ and $\theta = 2$ for $d > 3$.

The proof is based on Lemma 1 given below. The Lemma asserts that there exists a set K of n keys and a large enough set $B(T,n)$ of query parallelepipeds such that the subsets of K induced by members of $B(T,n)$ satisfy certain intersection conditions. The proof of Lemma 1 is given in Section 4.

Lemma 1. There is a set K of n keys and a set $B(T,n)$ of boxes satsfying the following properties.

1. $T\,|B(T,n)| = \Omega(n(\log_T n)^{d-\theta})$ where $\theta = 1$ for $d = 2,3$ and $\theta = 2$ for $d > 3$.
2. For distinct $b_1, b_2 \in B(T,n)$, $|b_1 \cap b_2 \cap K| < \frac{1}{T}\, min(|b_1 \cap K|, |b_2 \cap K|)$.

Using property 2 in Lemma 1, we show that for any H satisfying $P(H,T)$, we must have $|H| \geq |B(T,n)|$. Then Theorem 1 follows from the lower bound on $|B(T,n)|$ given by property 1 in Lemma 1. Let b_1, b_2 be distinct boxes in $B(T,n)$. As $b_1 \cap K$ is expressible as the union of at most T sets in H, there exists $h_1 \in H$ such that $|h_1| \geq (1/T)\,|b_1 \cap K|$ and $h_1 \subseteq (b_1 \cap K)$. Since $|(b_1 \cap K) \cap (b_2 \cap K)| < (1/T)\,|b_1 \cap K|$, h_1 cannot appear in the decomposition of $b_2 \cap K$ as the union of members of H. So with each b_i in $B(T,n)$ we can associate a distinct h_i in H.

2.2. Model B - Tree Model

In this model, the data structure is assumed to be a rooted tree. The set of tree vertices has a distinguished subset called data vertices, and each data vertex contains a data item which is an element in the commutative semigroup G. We let $data(v)$ denote the data value stored in a vertex v. With each edge in the tree is associated a condition, each condition being restricted to be a conjunction of binary comparisons. Given an input query in the form of a tuple of numbers, the query answering algorithm first visits the root. A vertex v is visited iff it is a son of some vertex u that has already been visited and the input tuple satisfies the condition associated with edge uv. We define $cond(v)$ to be the conjunction of conditions on the path from the root to vertex v. For any query region b, on being given the corresponding tuple $Q(b)$ as input, the query answering algorithm visits vertex v iff $Q(b)$ satisfies $cond(v)$. Finally, the response to a query is the semigroup sum of the data at all the visited data vertices.

For a fixed tree, the query time is said to be T if T is the maximum of the numbers of data vertices visited by the query answering algorithm for all possible queries. For a fixed set of records, the storage space S is defined to be the minimum number of data vertices that a corresponding tree data structure must have in order to guarantee a query time of T. We note that the cost of traversing edges and evaluating the associated conditions is not included in the query time, and that there is no bound on the degree of any vertex. The tree model allows the data structure to depend on the semigroup G and any correlation betwen the distributions of the given keys and the corresponding data values can be exploited to build a better data structure.

On the other hand, the tree nature of the data structure restricts the manner in which data locations may be accessed.

In this model, we consider two instances of orthogonal range query which we call the counting problem and the maximum problem. In the counting problem, the data value associated with each key is 1, and the response to a query is the number of keys located in the query box. The semigroup G is the semigroup on non-negative integers with the usual addition operation. For the maximum problem, G is the semigroup on non-negative integers with the addition operation defined by $x+y=\max\{x,y\}$. The response to a query is the maximum of all the data values whose keys are located in the query box.

The results for the counting problem, are summarized by the following theorem.

Theorem 2. In Model B, for the counting problem in a static database with n records, there is a space-time tradeoff $TS \geq \Omega(n(\log_T n)^{d-\theta})$ where $\theta=1$ for $d=2,3$ and $\theta=2$ for $d>3$.

The main lemma in the proof is stated below, we prove the lemma in Section 4.

Lemma 2. There is a set K of n keys and a set $B(T,n)$ of boxes satisfying the following properties.

1. $T\,|B(T,n)| = \Omega(n(\log_T n)^{d-\theta}$ where $\theta=1$ for $d=2,3$ and $\theta=2$ for $d>3$.
2. For distinct boxes b_1,b_2 in $B(T,n)$ and any vertex v, if both the input tuples $Q(b_1)$ and $Q(b_2)$ satisfy $cond(v)$ then there is a box b' such that $Q(b')$ also satisfies $cond(v)$ and $|b' \cap K| < \frac{1}{T}\,min(\,|b_1 \cap K|,\,|b_2 \cap K|\,)$.

Using property 2 in Lemma 2, we show that if the query time is T then the tree data structure must have at least $|B(T,n)|$ data vertices, and then the theorem will follow from the lower bound on $|B(T,n)|$ given by property 1 in Lemma 2. For any box b, since the query answering algorithm visits at most T data vertices with $Q(b)$ as input, there is a visited data vertex v such that $data(v) \geq (1/T)\,|b \cap K|$. Suppose for distinct boxes b_1,b_2 in $B(T,n)$, there is a data vertex u such that both $Q(b_1)$ and $Q(b_2)$ satisfy $cond(u)$ and $data(u) \geq (1/T)\,min(\,|b_1 \cap K|, |b_2 \cap K|\,)$. Then by Lemma 2 there is a box b' such that $Q(b')$ satisfies $cond(u)$ and $|b' \cap K| < data(u)$, so the query answering algorithm would work incorrectly with $Q(b')$ as input. So with each box b_i in $B(T,n)$ we can associate a distinct vertex u_i such that $Q(b_i)$ satisfies $cond(u_i)$ and $data(u_i) \geq (1/T)\,|b_i \cap K|$. So the tree must have at least $|B(T,n)|$ data vertices.

For the maximum problem, we assume that the data values associated with all the keys in K are distinct. We then have the following Theorem.

Theorem 3. In Model B, for the maximum problem in a static database with n records, we have a space-time tradeoff $(\log T + \log\log n)^{d-1}\, S \geq \Omega(n(\log n)^{d-\theta})$ where $\theta=1$ for $d=2,3$, and $\theta=2$ for $d>3$.

The main lemma in the proof is stated below, a proof of the lemma is given is Section 4.

Lemma 3. For the maximum problem, there is a set K of n keys such that there exists $K' \subseteq K$ satisfying the following properties.

1. $|K'| = \Omega(n/w(n))$ where $w(n)=1$ for $d=2,3,4$, and $w(n)=\log_2 n$ for $d>4$.
2. For each key $k \in K'$ there is a set of boxes $A(k)$ satisfying conditions 2.1, 2.2, 2.3 and 2.4.

2.1. The maximum data value in each box in $A(k)$ is $f(k)$.

2.2. $|A(k)| \geq 1$.

2.3. $\sum_{k \in K'} |A(k)| \geq c_2\, n(\log n)^{d-\theta}$ where $\theta=1$ for $d=2,3$, and $\theta=2$ for $d>3$, and c_2 is a constant dependent on d.

2.4. For each data vertex v and each key $k \in K'$, if $data(v)=f(k)$ then the number of boxes in $A(k)$ whose input tuples satisfy $cond(v)$ is at most δ where $\delta = (\log_2 T + (d+5)\log_2\log_2 n + d)^{d-1}$.

Once we have the above Lemma, we complete the proof as follows. For each box $b \in A(k)$, on being given $Q(b)$ as input, the query answering algorithm must visit a data vertex v such that $data(v)=f(k)$. So from Lemma 3, we can conclude that for each key k in K', the corresponding data value $f(k)$ is present in at least $\delta^{-1}\,|A(k)|$ data vertices in the tree, and since the data values associated with all the n keys are distinct, the number of data vertices is at least $\delta^{-1} \sum_{k \in K'} |A(k)|$ which is $\Omega(\delta^{-1}\, n(\log n)^{d-\theta})$.

2.3. Tree Model for the reporting problem

The tree model defined in Section 2.2 is not suited for the reporting problem because the degree of a vertex is unrestricted, and there is no cost for traversing edges. In such a model the reporting problem is solved optimally. Consider a tree with $n+1$ vertices, root of degree n, and a single distinct record in each leaf. The condition on the edge between a leaf and the root is a conjunction of comparisons which is true if and only if the query region contains the key whose record is in that leaf. Then any query visits only those data vertices which contain a record whose key lies in the query region.

So we restrict ourselves to trees where every vertex has degree less than some fixed constant (in fact letting the vertex degree be bounded by a slow growing function of n like $(\log n)^d$ is adequate). To compensate for the bounded degree we allow the condition associated with each edge to be a disjunction of comparisons. Thus $cond(v)$, the conjunction of the conditions on the path from the root to vertex v, is now a conjunction of disjunctions. Moreover, for each data vertex v, $data(v)$ is an unstructured set of records. We allow data vertices to share storage, so $data(v)$ is in effect the set of records acessed via data vertex v.

Consider a fixed set of records and a fixed tree for the set of records. Let $V(b)$ be the set of data vertices visited by the query corresponding to query region b. We only require that $\bigcup_{v \in V(b)} data(v)$ be a superset of the set of records whose keys lie in the query region b. So filtering search [2] is also included in this model. The time $T(b)$ required to answer the query corresponding to b is lower bounded by the number of vertices visited plus the size of $\bigcup_{v \in V(b)} data(v)$. As $T(b)$ is dependent on the output size it is not a correct measure of the overhead involved in answering the query corresponding to b, so we define a scaled query time T' as follows. Let $L(b)$ be the number of records whose keys are located in b. T' is said to be the scaled query time if

$$T' = \max_{b \text{ such that } 1 \le L(b) \le \log_2 n} \frac{T(b)}{L(b)}.$$

For a fixed set of records, the storage S is defined to be the minimum number of data vertices a corresponding tree must have to ensure a scaled query time of T'.

Theorem 4. In the tree model, for the reporting problem, there is space-time tradeoff $(\log T' + \log\log n)^{d-1}\; T'\, S \ge \Omega(n(\log n)^{d-\theta})$, where $\theta = 1$ for $d=2,3$, and $\theta=2$ for $d>3$.

The proof of Theorem 4 is lengthy and will appear in a detailed version of the paper. We also note that, in a tree model with restricted vertex degree and disjunctions of comparisons as conditions on edges, there are bounds similar to those in Theorem 4 for the maximum problem.

3. Canonical parallelepipeds and almost uniform distributions

We shall utilise a special class of parallelepipeds in obtaining the desired space-time tradeoffs. We assume that n is a power of 2 and let $I_l = \{[j2^l + 1, (j+1)2^l + 1) : 0 \le j < (n/2^l)\}$. I_l is the set of intervals obtained by breaking up $[1, n+1)$ into $n/2^l$ semi-closed intervals of equal size, each interval being closed on the left and open on the right. Let $I = I_0 \cup I_1 \cup \cdots \cup I_{\log_2 n}$. Then I^d is defined to be the set of canonical parallelepipeds, or equivalently canonical boxes.

For a box b, *dimensions*(b) defined to be the d-tuple $(i_1, i_2, \ldots, i_d)$ where i_j is the length of box b along the x_j-axis, for $1 \le j \le d$. We let $C(v)$ denote the set of all canonical boxes of volume v and $vol(b)$ the volume of a box b. Let $p(x)$ denote the smallest power of 2 greater than or equal to x. Let J be the canonical parallelepiped $J_0 \times J_1 \times \ldots \times J_{d-1}$ where for $0 \le i \le d-1$, $J_i = [2i(2p(d))^{-1} n + 1, (2i+1)(2p(d))^{-1} n + 1)$. The following lemmas list the properties of canonical boxes that we shall require.

Lemma 4. Let $b_1, b_2, \ldots, b_\delta$ be canonical boxes of volume v such that $\bigcap_{i=1}^{\delta} b_i \ne \phi$ and b the smallest box containing each of $b_1, b_2, \ldots, b_\delta$. Then b is a canonical box such that $vol(b) \ge v\, 2^{\mu-d}$, and $vol(\bigcap_{i=1}^{\delta} b_i) \le v\, 2^{d-\mu}$, where $\mu^{d-1} = \delta$.

Lemma 5. Let $v \ge n^{d-1}$, then $|C(v)| \le (v^{-1} n^d (d \log_2 n - \log_2 v + d)^{d-1})$.

We define $E(V(T,n), r(T))$ to be a largest set of boxes satisfying the following conditions.

1. Each box in $E(V(T,n), r(T))$ is a subbox of the canonical box J.
2. Each box in $E(V(T,n), r(T))$ is a canonical box of volume $V(T,n)$ and the intersection of any two boxes in $E(V(T,n), r(T))$ is a canonical box of volume at most $(V(T,n)/r(T))$.
3. For each box b in $E(V(T,n), r(T))$, each dimension of b is at least $r(T)$, and every canonical box b' such that *dimensions*$(b') =$ *dimensions*(b) and $b' \subseteq J$ is also present in $E(V(T,n), r(T))$.

Lemma 6. Let $V(T,n)$ and $r(T)$ take on values that are powers of 2, and let $V(T,n) = o(n^{d-1+\epsilon})$ and $r(T) = o(n^{d-1+\epsilon})$ for small fixed ϵ. Then the number of possibilities for the dimensions of a box in $E(V(T,n), r(T))$ is $\Omega((\log n/\log(r(T)))^{d-1})$, and the number of boxes in $E(V(T,n), r(T))$ that have identical dimensions is $\Omega(n^d/V(T,n))$.

Having described canonical boxes, we shall proceed to almost uniform distributions of n keys, the distributions are termed almost uniform because the number of keys in each canonical box does not deviate too far from the volume of the canonical box divided by n^{d-1}. For $d=2,3$, we can explicitly construct such distributions and thereby get Theorem 5. For $d>3$, we have to resort to counting arguments and show that the number of distributions of n keys which do not satisfy the properties in Theorem 6, is less than the total of n^{dn} possible distributions.

Theorem 5. For $d=2,3$, there is a set K of n keys such that for each canonical box b,

$$\left\lfloor \frac{vol(b)}{n^{d-1}} \right\rfloor \le |b \cap K| \le \left\lceil \frac{vol(b)}{n^{d-1}} \right\rceil.$$

We shall briefly outline how to construct a set K of n keys satisfying the conditions in Theorem 5 for $d=2$. We use an inductive construction. Let x_1 and x_2 denote the two attributes of a key. Suppose we have a such a set for $n=m$. We make two copies of the set of keys. In the first copy transform x_1 to $2x_1$-1 and x_2 to x_2. In the second copy transform x_1 to $2x_1$ and x_2 to x_2+m. A canonical box of volume $2m$ and x_1 dimension equal to 1 contains exactly one key, as each key has a distinct value for x_1. If $[x_{11}, x_{12}) \times [x_{21}, x_{22})$ is a canonical box of volume m corresponding to $n=m$, then $[2x_{11}-1, 2x_{12}-1) \times [x_{21}, x_{22})$ and $[2x_{11}-1, 2x_{12}-1) \times [x_{21}+m, x_{22}+m)$ are canonical boxes of volume $2m$ corresponding to $n=2m$. Moreover, all canonical boxes, of volume $2m$ and x_1 dimension greater than or equal to 2, corresponding to $n=2m$, can be obtained in this manner, and then by the induction hypothesis each of them contains exactly one key from one of the two copies.

Theorem 6. Let σn^{dn} be the number of distinct sets K of n keys, each key in N^d, which satisfy the three properties given below. Then σ tends to 1 as n tends to ∞ and $\sigma = (1 - o(1/n))$.

1. Let $a(n) = 2p(\log_2 n)$ and let $v_0 = a(n) n^{d-1}$. For each canonical box b,

$$\left\lfloor \frac{vol(b)}{v_0} \right\rfloor \le |b \cap K| \le 6\, a(n) \left\lceil \frac{vol(b)}{v_0} \right\rceil.$$

2. Each canonical box of volume n^{d-1} contains at most $\log_2 n$ keys.
3. $|J \cap K| \ge (n/(4(2p(d))^{2d}))$.

4. Proofs of Lemmas

In this section we give proofs of the main lemmas in Theorems 1, 2 and 3. For $d = 2,3$, let K be a set of n keys as specified by Theorem 5 and for $d > 3$ let K be as specified by Theorem 6. In Model B, as we restrict ourselves to binary comparisons, the only possible comparisons are those between two elements in the input tuple, and those between an element in the input tuple and a constant. We shall focus on canonical boxes that are subboxes of the canonical box J. For each box $b \subseteq J$, the input tuple $Q(b) = (x_{11}, x_{12}, x_{21}, x_{22}, \ldots, x_{d1}, x_{d2})$ is such that $x_{11} < x_{12} < x_{21} < x_{22} < \ldots. < x_{d1} < x_{d2}$, and so a comparison involving two elements in the input tuple has the same outcome for each subbox b of J. Then, in Model B, we need to analyze only comparisons between a single element in the input tuple and a constant. We note that, if the input tuple satisfies a comparison between x_j, the $(j)^{th}$ element in the tuple, and a constant when x_j takes on the value α_1 as well as when x_j takes on the value α_2, then the input tuple satisfies the comparison whenever x_j takes on any value between α_1 and α_2.

Let $r(T) = (64\,T(2p(d))^{2d})$, let $V_1(T,n) = (r(T)n^{d-1})$ and let $V_2(T,n) = (r(T)a(n)n^{d-1})$. For $d = 2,3$, we let $B(T,n) = E(V_1(T,n), r(T))$ and for $d > 3$ we let $B(T,n) = \{b : b \in E(V_2(T,n), r(T)), |b \cap K| \geq 6\,T\,a(n)\}$.

Proof of Lemma 1. We sketch the proof for $d > 3$, for $d = 2,3$, the reasoning is similar. Using the information that the canonical box J contains at least $(n/(4(2p(d))^{2d}))$ points, one can show that the number of boxes in $B(T,n)$ that have identical dimensions is $\Omega(n/(r(T)a(n)))$, and then from Lemma 6 we have $T\,|B(T,n)| = \Omega(n(\log n)^{d-2})$. The intersection of any two distinct boxes b_1, b_2 in $B(T,n)$ is a canonical box of volume at most $(a(n)n^{d-1})$ and so from Theorem 6 $|b_1 \cap b_2 \cap K| < 6\,a(n) < (1/T)\,min(|b_1 \cap K|, |b_2 \cap K|)$.

Proof of Lemma 2. The lower bound on $|B(T,n)|$ follows from Lemma 1. We only consider the case $d > 3$, the reasoning in the case $d = 2,3$, is almost identical. Suppose there are distinct boxes b_1, b_2 in $B(T,n)$ and a vertex v such that both the input tuples $Q(b_1)$ and $Q(b_2)$ satisfy $cond(v)$. We show that there exists a box b' such that $|b' \cap K| < 6\,a(n) < (1/T)\,min(|b_1 \cap K|, |b_2 \cap K|)$ and $Q(b')$ satisfies $cond(v)$. There are two cases.
Case 1. $b_1 \cap b_2 \neq \phi$, then $Q(b_1 \cap b_2)$ satisfies $cond(v)$ and $|b_1 \cap b_2 \cap K| < 6\,a(n)$.
Case 2. $b_1 \cap b_2 = \phi$, let $Q(b_1) = (\alpha_{11}, \alpha_{12}, \ldots., \alpha_{d1}, \alpha_{d2})$ and $Q(b_2) = (\beta_{11}, \beta_{12}, \ldots., \beta_{d1}, \beta_{d2})$. Then for some i, $\alpha_{i1} < \alpha_{i2} \leq \beta_{i1} < \beta_{i2}$ or $\beta_{i1} < \beta_{i2} \leq \alpha_{i1} < \alpha_{i2}$. Let $Q(b_1')$ be obtained from $Q(b_1)$ by replacing α_{i2}, the $(2i)^{th}$ element in $Q(b_1)$, by $(\alpha_{i1} + 1)$, and similarly, let $Q(b_2')$ be obtained from $Q(b_2)$ by replacing β_{i2}, the $(2i)^{th}$ element in $Q(b_2)$, by $(\beta_{i1} + 1)$. Then one of $Q(b_1')$, $Q(b_2')$ satisfies $cond(v)$. As each dimension of b_1 and b_2 is at least $r(T)$, b_1' and b_2' are canonical boxes of volume at most $a(n)n^{d-1}$ and so $|b_1' \cap K| < 6\,a(n)$ and $|b_2' \cap K| < 6\,a(n)$.

Proof of Lemma 3. Let the data values associated with all the keys in K be distinct. Let $maxin(b)$ denote the maximum of all the data values whose keys lie in the box b. For each key $k \in K$, let

$$A(k) = \{b : b \in C(n^{d-1}), b \subseteq J, maxin(b) = f(k)\}.$$

We have the following lemma.

Lemma 7. There exists $K_1 \subseteq K$ such that

1. $|K_1| \geq c_3(n/w(n))$ where c_3 is a constant dependent on d, $w(n) = 1$ for $d = 2,3,4$, and $w(n) = \log_2 n$ for $d > 4$.
2. For each $k \in K_1$, $|A(k)| \geq 1$.
3. $\sum_{k \in K_1} |A(k)| = c_2 n(\log n)^{d-\theta}$ where $\theta = 1$ for $d = 2,3$, and $\theta = 2$ for $d > 3$, and c_2 is a constant dependent on d.

Let k be a key in K_1 such that for some vertex $u(k)$ satisfying $data(u(k)) = f(k)$, there are δ boxes $b_1(k), b_2(k), \ldots., b_\delta(k)$ in $A(k)$ such that each of $Q(b_1(k)), Q(b_2(k)), \ldots., Q(b_\delta(k))$ satisfies $cond(u(k))$, where $\delta = (\log_2 T + (d+5)\log_2\log_2 n + d)^{d-1})$. Let K_2 be the set of all such keys in K_1. For each key k in K_2, let $g(k)$ be the smallest canonical box containing each of $b_1(k), b_2(k), \ldots., b_\delta(k)$. Then by Lemma 4, $vol(g(k)) \geq (n^{d-1}\,T(\log_2 n)^{d+5})$ and $Q(g(k))$ satisfies $cond(u(k))$. From Lemma 5, the are at most $O(n\,T^{-1}(\log n)^{-5})$ possibilities for $g(k)$. Suppose $|K_2| = \Omega(n/(w(n)(\log n)^2))$. Then we can find $T+1$ keys $k_1, k_2, \ldots., k_{T+1}$ such that $g(k_1) = g(k_2) = \ldots. = g(k_{T+1})$ and $g(k_1)$ satisfies each of $cond(u(k_1))$, $cond(u(k_2))$, ..., $cond(u(k_{T+1}))$ for distinct vertices $u(k_1)$, $u(k_2)$, ..., $u(k_{T+1})$. The query time then exceeds T. Thus $|K_2| = o(n/(w(n)(\log n)^2))$, and $\sum_{k \in K_2} |A(k)| = o(n(\log n)^{d-3})$. We let $K' = (K_1 - K_2)$.

5. Conclusion

We have obtained space-time tradeoffs for orthogonal range query in two models, the arithmetic model and the tree model. Most data structures used in practice are rooted trees and so it may be worth studying more problems in the context of Model B. We conclude by raising questions related to the tree model.

1. Drawing an analog with decision trees, what happens when the conditions associated with tree edges are allowed to be comparisons involving linear or higher order poylnomial functions of the input ? Do the bounds weaken in such a situation ?
2. What kind of bounds can one obtain for queries other than orthogonal range query, say circular range query or polyhedral query ?
3. Can the bounds obtained for the tree model be extended to data structures which can be modelled as directed acyclic graphs?

Acknowledgements

The author would like to thank Prof. C. L. Liu and S. Kapoor for helpful discussions and suggestions.

References

[1] J. L. Bentley and H. A. Maurer, Efficient worst-case data structures for range searching, *Acta Informatica*, 13, 1980, pp. 155-168.

[2] B. Chazelle, Filtering Search: A new approach to query answering, *Proc. 24th Annual IEEE Symp. Found. Comp. Sci.*, 1983, pp. 122-132.

[3] R. Cole and C. K. Yap, Geometric Retrieval Problems, *Proc. 24th Annual IEEE Symp. Found. Comp. Sci.*, 1983, pp. 112-121.

[4] Fredman M. L., The inherent complexity of dynamic data structures which accomodate range queries, *Proc. 21st Annual IEEE Symp. Found. Comp. Sci.*, 1980, pp. 191-200.

[5] Fredman M. L., A lower bound on the complexity of orthogonal range queries, *JACM*, Vol. 28, No. 4, 1981, pp. 696-705.

[6] Fredman M. L., Lower bounds on the complexity of some optimal data data structures, *SIAM J. Comput.*, Vol. 10, No. 1, 1981, pp. 1-10.

[7] H. Gabow, J. Bentley, and R. Tarjan, Scaling and Related Techniques for Geometric Problems, *Proc. 16th Annual Symp. Theory of Comput.*, 1984, pp. 135-143.

[8] G. S. Lueker, A data structure for orthogonal range queries, *Proc. 19th Annual IEEE Symp. Found. of Comp. Sci.*, 1978, pp. 28-34.

[9] D. E. Willard, New data structures for orthogonal range queries, Tech. Report TR-22-78, Aiken Computer Lab., 1978, Harvard University.

[10] D. E. Willard, Polygon Retrieval, *SIAM J. Comput.*, Vol. 11, 1982, pp. 149-165.

[11] A. C. Yao, Space-Time tradeoff for answering range queries, *Proc. 14th Annual Symp. Theory of Comput.*, 1982, pp. 128-136.