Space-Time Tradeoff for Answering Range Queries*
(Extended Abstract)

Andrew C. Yao
Computer Science Division
Department of EECS
University of California
Berkeley, California 94720

1. INTRODUCTION

Consider a database 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 keys, the database system is expected to return the set of records, or a certain function of the set of records, whose keys satisfy all the constraints. For example, consider a geographic database in which the record for a city contains among other items its location (as the key) and its population (as a data field). A range query may ask for the total population of cities whose locations are within r miles of a certain point; another possibile range query may ask for the number of cities in a certain region. There is an extensive literature on efficient algorithms for handling various types of range queries (see, e.g. Bentley and Mauer [1], Leuker [7], Rivest [8], Willard [11],[12]). As in all data structure problems, the optimality question (i.e., whether one has found the best possible solution) is much harder to answer. Recently, an interesting model was developed by Fredman [3-6] to discuss the inherent complexity of processing a sequence of m INSERT, DELETE, and QUERY instructions for range queries. Several elegant results were obtained which offered insight into questions such as "why a sequence of circular range queries seems much more difficult to process than a sequence of orthogonal queries," and "why a sequence of $O(n)$ instructions for d-dimensional orthogonal range queries takes $\Omega(n(\log n)^d)$ time to process."

In this paper, we raise and investigate the question of (storage) space- (retrieval) time tradeoff for a static database, in the general framework of Fredman's. As will be seen, such tradeoff results also lead to lower bounds on the complexity of processing a sequence of m INSERT and QUERY instructions. The latter results are incomparable to Fredman's, since the presence of DELETE instructions was crucial for his proof technique. We will present our results in detail in the next few sections. Here we will only mention three main conclusions. Firstly, circular query is shown to be intrinsically hard in the sense that, for some static database with n records, there is a space-time tradeoff $TS > n^{1+\varepsilon}$ where $\varepsilon>0$; in contrast, orthogonal query can always be implemented with space $S=O(n(\log n)^k)$ and time $T=O((\log n)^k)$ for fixed k. Furthermore, any algorithm for processing $O(n)$ INSERT and QUERY instructions must use time $\Omega(n^{1+\varepsilon})$ in the worst case. Secondly, for the "interval" query, we have determined the space-time tradeoff quite precisely to be $T \approx \alpha(S,n)$, where α is the inverse to an Ackermann's function first defined by Tarjan [9]. This is a rare case where the function α arises outside the context of path compression, and is obtained through a totally independent derivation. Thirdly, we prove that, for the interval query, any algorithm to process a sequence of $O(n)$ INSERT and QUERY must take time $\Omega((n \log n)/(\log \log n))$ in the worst case. This means that one cannot hope to maintain the most efficient static data structure (with retrieval time $\alpha(S,n)$) in the dynamic case.

We remark that the space and time requirements have traditionally been used as performance measure for range query algorithms, and different algorithms exhibiting space-time tradeoff have been proposed in the literature [1,2,7,11,12].

The proofs of the theorems are lengthy. In this extended abstract, we only present the proofs for Theorems 1 and 2. The proofs for the other theorems can be found in [13] and [14].

2. INTERVAL QUERY

Let $X = (x_1,x_2,\ldots,x_n)$ be an n-tuple of real numbers. We wish to store information in m

*This work was done while the author was visiting the Computer Science Department, IBM San Jose Research Center, San Jose, California.

memory cells, such that every interval query "q_{ij} = ?" can be answered quickly, where $q_{ij} = x_i + x_{i+1} + \ldots + x_j$. Let $z_1, z_2, \ldots, z_m$ denote the contents of the memory cells. One solution is to set $z_i \leftarrow x_1 + x_2 + \ldots + x_i$ for all i; clearly each query "q_{ij} = ?" can be answered by fetching z_{i-1} and z_j and then computing $z_j - z_{i-1}$. This solution is very economical, since it uses minimum space n and achieves an $O(1)$ retrieval time. Note that, however, a subtraction is needed to answer a query. In this paper, we are interested in solutions that do not use subtractions. (See the remark at the end of this section, for why we are interested in such solutions.)

A scheme for X is a set of linear functions $R = \{z_i = \Sigma \lambda_{ij} x_j \mid 1 \leq i \leq m'\}$ with $\lambda_{ij} \geq 0$. We call R a (t,m)-scheme if $m' \leq m$ and for any $1 \leq i \leq j \leq n$, q_{ij} can be written as $\Sigma_{k \in T} \mu_k z_k$ for some $\mu_k \geq 0$, $|T| \leq t$; we will say that T uses (at most) space m and retrieval time t. Let $f(m,n)$ be the minimum t for which there exists a (t,m)-scheme for $(x_1, x_2, \ldots, x_n)$. Clearly, $f(m,n)$ is finite only if $m \geq n$. The main result of this section is the following theorem.

Theorem 1. For $m \geq n \geq 1$,

$$f(m,n) = \Theta\left(\alpha(m,n) + \frac{n}{m-n+1}\right)$$

The term $n/(m-n+1)$ is negligible unless m is very close to n. The function α is the inverse to a variant of Ackermann's function, as defined in Tarjan [9]. It is a very slowly-varying function; for example $\alpha(2n,n) = O(\log^* \log^* n)$. This is a rare case where α occurs in the analysis of algorithms outside the context of path compression [9]. (Another case was given in [10], where the problem was solved by a reduction to path compression, however.)

In the above model, the retrieval time t counts basically the number of arithmetic operations needed to answer a query, while ignoring the time needed to find the proper memory cells. Such "arithmetic" models for query processing were first considered by Fredman [3-6]. As in Fredman's papers, one can discuss the present query problem for general semigroups. Let G be any commutative semigroup with an addition operation "+". A (t,m)-scheme for a list of n items $(x_1, x_2, \ldots, x_n)$ of G is a set of $m' \leq m$ functions $z_i = \Sigma_j \lambda_{ij} x_j$ ($1 \leq i \leq m'$, λ_{ij} nonnegative integers), such that any q_{ij} can be written in the form $\Sigma_{1 \leq k \leq t} \mu_k z_{\ell k}$ for some integral $\mu_k \geq 0$. We define $f_G(m,n)$ to be the minimum retrieval time t for any scheme using space m. The same technique for proving Theorem 1 also gives the same bound for a general class of semigroups. Let us call a semigroup S faithful if, for every integral $\delta_i, \delta'_j > 0$,

$$\sum_{i \in T_1} \delta_i x_i = \sum_{j \in T_2} \delta'_j x_j$$

cannot be an identity for $x_1, x_2, \ldots, x_n \in G$ unless $T_1 = T_2$. For example, as can be easily verified, the set of real numbers with "max{x,y}" as the addition operation is a faithful semigroup; so is the set $\{0,1\}$ with the logical "OR" being the addition operation.

Theorem 2. Let G ve a faithful commutative semigroup. Then for $m \geq n \geq 1$

$$f_G(m,n) = \Theta\left(\alpha(m,n) + \frac{n}{m-n+1}\right).$$

We remark that the constants in the Θ-notation in Theorem 2 are the same as in Theorem 1 (hence independent of G).

Remarks. We are interested in solutions without subtractions, because such solutions correspond to storage schemes when the data space forms a semigroup. In many applications, the addition operation does not have an inverse (i.e., no subtraction available). For example, suppose we are interested in the largest city and its population in the region specified by the range query, then the operation "x + y" is in fact "max{x,y}", and max has no inverse.

3. CIRCULAR QUERY

Let G be a faithful commutative semigroup with an addition operation "+". Let Z be the set of all pairs (i,j) of integers. A record (k,x) is a pair of key $k \in Z$ and datum $x \in G$. A file f is a finite collection of records. A circular query $q = \text{QUERY}(a,r)$ is specified by a center $a \in Z$ and a radius $r > 0$. For a file F, the response, resp$(a,r;F)$, to a circular query QUERY(a,r) is defined to be the sum of data x whose key k satisfy $|k-a| < r$.

Let F be a given file. A storage scheme R for F is a family of linear functions (with nonnegative coefficient) $\{z_1, z_2, \ldots, z_S\}$ such that, for any $a \in Z$ and $r > 0$, there exists an identity

$$\text{resp}(a,r;F) = \Sigma_{j \in V} \mu_j z_j \qquad (1)$$

valid for all values of $x_i \in G$, for some integers $\mu_j > 0$ and $V \subseteq \{1,2,\ldots,S\}$. Let $t_F(a,r;R)$ be the

minimum $|V|$ for which resp(a,r;F) can be written in the above form; let $t_F(R) = \max_{a,r} t_F(a,r;R)$. We say that R uses <u>space</u> X and <u>time</u> $T = t_F(R)$.

We emphasize that the keys k_i in F are considered fixed, and identity (1) need only be valid for this set of keys. However, the identity must be true for all possible values of $x_i \in G$.

We will now give a theorem which gives a space-time tradeoff for the file F, whose keys consist of the lattice points in the $\sqrt{n} \times \sqrt{n}$ square. Let $F_n = \{(k_{ij},x_{ij}) | 0 \le i,j < \sqrt{n}\}$, where $k_{ij} = (i,j)$.

<u>Theorem 3</u>. Let R be a storage scheme for the file F_n. If R uses space S, then there exist some integral $a = (i,j)$ and r with $0 \le i,j, r < \sqrt{n}$ such that

$$t_F(a,r;R) > n^{1+\varepsilon}/S,$$

where $\varepsilon > 0$ is a fixed constant.

<u>Corollary</u>. Any storage scheme for F_n must satisfy the time-space-time tradeoff $TS > n^{1+\varepsilon}$.

Note that the query QUERY(a,r) with a high retrieval time in Theorem 3 is of a special type, i.e., the center a is in the $\sqrt{n} \times \sqrt{n}$ square. Thus, Theorem implies that an efficient data structure cannot be designed for F_n to answering circular queries, even if we further restrict the queries to this special type.

4. DYNAMIC QUERIES

In this section we consider the question of processing a sequence of record insertions and queries. Essentially, we are looking at the dynamic range query model of Fredman [6], except that the deletion of records from the file is not allowed. However, the proof technique used in [6] cannot be applied here to yield lower bounds on the complexity for our problem, because the presence of the deletion instruction is crucial to that technique.

We will develop techniques, which allow us to derive lower bounds to the complexity in this dynamic setting, from the static space-time constraint for similar types of queries. Theorem 4 below is obtained from Theorem 3 in this fashion. Theorem 5 is also obtained along this line. Its proof, however, is somewhat more involved (it is not derived from Theorem 1 or 2).

Let us describe briefly the mathematical model. For definiteness, assume that we are interested in circular queries (the description is valid for any type of range queries with the obvious modifications). Initially, the file F is empty. We wish to process a sequence σ of instructions $\alpha_1,\alpha_2,\ldots,\alpha_p$, where each α_i is of either one of the two types: INSERT(k,x), QUERY(a,r). The instruction INSERT(k,x) requires that $F \leftarrow F \cup \{(k,x)\}$, and QUERY(a,r) asks that resp(a,r;F) be returned for the current F.

In the model we are considering, there is an infinite array of variables $z_1,z_2,\ldots$. An algorithm A specifies how the instructions are to be implemented using these variables. To process an instruction α_i = INSERT(k,x) one carries out a sequence of operations $\beta_1,\beta_2,\ldots,\beta_w$, where each β_u is either of the form $z_j \leftarrow x$ or $z_j \leftarrow \lambda z_k + \mu z_k$, ($\lambda,\mu$ nonnegative integers). To process an instruction α_i = QUERY(a,r), one carries out $\beta_1,\beta_2,\ldots,\beta_v$, where each β_u is either of the form $z_j \leftarrow \lambda z_k + \lambda' z_k$, ($\lambda,\lambda'$ nonnegative integers), <u>return</u>(z_j), or <u>return</u>($\emptyset$). The algorithms are fully adaptive; the number of operations and the choice of the operations used to process an instruction can depend on the key values of the queries that have been processed so far. Note that QUERY(a,r) takes at least $t_F(a,r;R)$ operations to process, where F is the current file and R is the collection of contents of the variables used (one operation for <u>return</u>(z_j) and $t_F(a,r;R)$-1 to get z_j). The cost $C(\sigma;A)$ is the total number of operations used to process the sequence σ.

Let $\varepsilon' = \varepsilon/2$, where $\varepsilon > 0$ is the constant in Theorem 3. Assume that the semigroup G under consideration is faithful.

<u>Theorem 4</u>. For any algorithm A for processing dynamic circular queries, there exists a sequence σ of 2n INSERT and (circular) QUERY instructions such that $C(\sigma,A) = \Omega(n^{1+\varepsilon'})$.

<u>Theorem 5</u>. For any algorithm A for processing dynamic interval queries, there exists a sequence σ of 2n INSERT and (interval) QUERY such that $C(\sigma,A) = \Omega(n \log n/\log\log n)$.

The above two theorems extend two analogous results of Fredman's [4] [6], where deletions of records may occur. The constant ε' we obtained is small. Recently, Fredman had shown (private communication) that ε' can be taken to be 1/4 in Theorem 4.

It is worth mentioning that the following result is obtained in the proof of Theorem 5, and is of independent interest.

<u>Theorem 6</u>. For the (2-dimensional) orthogonal query, there is a file F such that any storage scheme must satisfy the space-time constraint

$$T \log (S \log n/n) = \Omega(\log n).$$

For example, if $S = O(n)$, then $T = \Omega(\log n/\log\log n)$.

5. PROOF OF THEOREM 1

5.1 Ackermann's Functions.

In this section, we discuss some properties of $\alpha(m,n)$ and related functions. Most of the

proofs use straightforward induction, and will be omitted.

Let $A(i,j)$ be a function defined on all integers $i,j \geq 0$ as follows:

$$A(0,j) = 2j \text{ for } j \geq 0, \quad (2)$$
$$A(i,0) = 0,\ A(i,1)=2 \text{ for } i \geq 1,$$
$$A(i,j) = A(i-1,A(i,j-1)) \text{ for } i \geq 1,\ j \geq 2.$$

As in Tarjan [9], we define α by

$$\alpha(m,n)=\min\{i \mid i\geq 1,\ A(i,4\lceil m/n\rceil) > \log_2 n\}. \quad (3)$$

For any real number x, let

$$a(x,j)=\min\{i \mid i\geq 1,\ A(i,j) > x\}. \quad (4)$$

Then

$$\alpha(m,n) = a(\log_2 n,\ 4\lceil m/n\rceil). \quad (5)$$

A function $B(i,j)$, defined for all integers $i,j \geq 0$, is said to be monotone if it is nondecreasing in both variables, i.e. $B(i,j+1) \geq B(i,j)$ nad $B(i+1,j) \geq B(i,j)$ for all $i,j \geq 0$.

Lemma 1. The function $A(i,j)$ is monotone.

Proof. This can be proved easily by induction. See formulas (7) - (10) in [9]. □

Lemma 2. $a(x,j) = \Theta(a(x,j'))$ if $j,j' \geq 4$ and $\frac{1}{100} \leq \left|\frac{j}{j'}\right| \leq 100$.

Lemma 3. $a(x,j) = O(a(\log_2 x,j))$ if $x \geq 1$, $j \geq 4$.

We will also need the function $H(i,j)$ defined for all integers $i,j \geq 0$ as follows:

$$H(0,j)=12j^3 \text{ for } j \geq 0, \quad (6)$$
$$H(i,0)=0\ \ H(i,1) = 12 \text{ for } i \geq 1,$$
$$H(i,j)=H(i-1,H(i,j-1)) \text{ for } i \geq 1,\ j \geq 2.$$

Lemma 4. The function $H(i,j)$ is monotone.

Lemma 5. $H(i,j) < A(i+4,j+4)$ for all $i,j \geq 0$.

5.2 Upper Bound.

We will show that

$$f(m,n) = O\left(\alpha(m,n) + \frac{n}{m-n+1}\right). \quad (7)$$

A. The Case $m \geq 2On$.

In this case it suffices to prove that

$$f(m,n) = O(\alpha(m,n)). \quad (8)$$

For each integer $i,j \geq 0$, let $D(i,j)$ be the largest integer n_0 such that, for every $n \leq n_0$, there exists a $(2(i+1),4(j+1)n)$-scheme for n items. It is easy to see that, for all $m \geq 4n$,

$$f(m,n)\leq \min\{2(i+1) \mid D(i,\lfloor m/(4n)\rfloor-1)\geq n\ . \quad (9)$$

We remark that the right-hand side of the inequality may be ∞. We will derive an upper bound on $D(i,j)$ and use inequality (9) to prove equation (8). Note that $D(i,j)$ is clearly monotone by its definition.

Lemma 6.

$$D(0,j)\geq 2^{4j+3} \text{ for } j\geq 0,$$
$$D(i,0)\geq 8,\ D(i,1)\geq 128 \text{ for } i\geq 1.$$

Proof. We first show that, for any $n \geq 1$, there exists a $(2,(1+\lceil\log_2 n\rceil)n)$-scheme for n items. This is obviously true for $n = 1,2$. Inductively, let $\ell > 1$, and suppose the statement is true for all $1 \leq n \leq 2^{\ell-1}$; we will prove it for all $2^{\ell-1} < n \leq 2^{\ell}$.

Let $2^{\ell-1} < n \leq 2^{\ell}$ and $X = (x_1,x_2,\ldots,x_n)$. Consider the following scheme $S = \{z_1,z_2,\ldots,z_m\}$, where

$$z_s=x_s+x_{s+1}+\ldots+x_{\lfloor n/2\rfloor} \text{ for } 1\leq x\leq \lfloor n/2\rfloor,$$
$$z_s=x_{\lfloor n/2\rfloor+1}+\ldots+x_{s-1}+x_s \text{ for } \lfloor n/2\rfloor<s\leq n,$$

and

$$\{z_{n+1},z_{n+2},\ldots,z_m\}=S_1\cup S_2 \text{ with } S_1 \text{ being a}$$
$$(2,(1+(\ell-1))\lfloor n/2\rfloor)\text{-scheme}$$
for $(x_1,x_2,\ldots,x_{\lfloor n/2\rfloor})$, S_2 being a
$$(2,(1+(\ell-1))\lceil n/2\rceil)\text{-scheme}$$
for $(x_{\lfloor n/2\rfloor+1},\ldots,x_n)$,

Clearly, this is a $(2,m)$-scheme with $m=n+(1+(\ell-1))n=(1+\lceil\log_2 n\rceil)n$. This completes the inductive step.

For $n \leq 2^{4j+3}$, we have

$$4(j+1)n \geq (1+\lceil\log_2 n\rceil)n;$$

the previous discussion implies that there exists a $(2,4(j+1)n)$-scheme. This means $D(0,j)\geq 2^{4j+3}$ for $j \geq 0$. As $D(i,j)$ is monotone, we have for $i \geq 1$

$$D(i,0) \geq D(0,0) \geq 8, \text{ and}$$
$$D(i,1) \geq D(0,1) \geq 128. \quad \square$$

Remark. $D(i,j) \geq 8$ for all $i,j \geq 0$ because of the monotonicity of the function D.

Lemma 7. For $i \geq 1$, $j \geq 2$,

$$D(i,j) \geq D(i,j-1)D(i-1,\lfloor\tfrac{1}{8}D(i,j-1)\rfloor).$$

Proof. We prove by induction on the lexicographic order of (i,j). Let $i \geq 1$, $j \geq 2$, and assume that we have proved the statement for all (i',j') that are lexicographically less than (i,j). Let $1 \leq n \leq D(i,j-1)D(i-1,\lfloor\frac{1}{8}D(i,j-1)\rfloor)$. We will construct a $(2(i+1),4(j+1)n)$-scheme for a list of n items $(x_1,x_2,\ldots,x_n)$.

Without loss of generality, we assume that

$$D(i,j-1) < n \le D(i,j-1)D(i-1,\lfloor\tfrac{1}{8}D(i,j-1)\rfloor). \quad (10)$$

Let $k = D(i,j-1) \ge 8$, $\ell = \lceil n/k \rceil \ge 2$. Partition $X = (x_1,x_2,\ldots,x_n)$ into ℓ consecutive blocks $T_1,T_2,\ldots,T_\ell$, where

$$T_v = (x_s \mid (v-1)k<s\le vk) \text{ for } 1 \le v < \ell,$$
$$T_\ell = (x_s \mid (\ell-1)k<s\le n).$$

We now construct a scheme S for X, consisting of three types of members.

Type 1: For each $T_v (1 \le v \le \ell)$, construct a $(2(i+1),4j|T_v|)$-scheme. This is possible since $|T_v| \le k = D(i,j-1)$. Let S_1 be the union of these schemes. Clearly, $|S_1| \le 4jn$.

Type 2: Let $S_2 = \{z_{v,u}, z'_{v,u} \mid 1 \le v \le \ell, 1 \le u \le |T_v|\}$, where

$$z_{v,u} = \sum_{0<s<u} x_{(v-1)k+s},$$

$$z'_{v,u} = \sum_{u\le s\le |T_v|} x_{(v-1)k+s}.$$

Clearly, $|S_2| = 2n$.

Type 3: As $\ell \le D(i-1,\lfloor\frac{1}{8}D(i,j-1)\rfloor)$, we can construct a $(2i,4(\lfloor\frac{1}{8}D(i,j-1)\rfloor+1)\ell)$-scheme τ for a list of ℓ items $(y_1,y_2,\ldots,y_\ell)$. Let $S_3 = \{I(w) \mid w \in \tau\}$, where $I(w)$ is the expression obtained from w by substituting each occurrence of y_v by* $\sum_{x_s\in T_v} x_s$. Using the facts $k \ge 8$ and $\ell \ge 2$, we have

$$|S_3| = |\tau|$$
$$= 4(\lfloor\tfrac{1}{8}D(i,j-1)\rfloor+1)\ell$$
$$\le 4\cdot\tfrac{1}{4}D(i,j-1)\cdot 2\frac{n}{D(i,j-1)}$$
$$= 2n.$$

Thus, the scheme S uses space

$$|S| \le 4jn + 2n + 2n$$
$$= 4(j+1)n.$$

*Although T_v is not a set but a list, we will use the same notation T_v for the set of items in the list; the meaning will be clear from the context.

To complete the proof, we must show that any query $q_{a,b}=x_a+x_{a+1}+\ldots+x_b$ can be written as the linear combination (with positive coefficients) of at most $2(t+1)$ members of S. We distinguish two cases:

Case a. $x_a,x_b \in T_v$ for some v: Then $q_{a,b}$ is the linear combination of at most $2(i+1)$ members of S_1 by the definition of S_1.

Case b. $x_a \in T_r$, $x_b \in T_{r'}$ with $r < r'$: Choose $p \le 2i$ members $I(w_1),I(w_2),\ldots,I(w_p)$ from S_3 such that

$$y_{r+1}+y_{r+2}+\ldots+y_{r'-1} = \sum_{1\le d\le p} \lambda_d w_d \quad \text{with} \quad \lambda_d \ge 0.$$

This implies

$$\sum_{rk<s\le(r'-1)k} x_s = \sum_{1\le d\le p} \lambda_d I(w_d). \quad (11)$$

Let $z'_{r,s}$ and $z_{r',s'}$ be members of S_2 such that

$$x_a + x_{a+1} +\ldots+ x_{rk} = z'_{r,s}, \text{ and}$$
$$x_{(r'-1)k+1} +\ldots+ x_{b-1} + x_b = z_{r',s'}. \quad (12)$$

Then $q_{a,b} = z'_{r,s} + z_{r',s'} + \sum_{1\le d\le p}\lambda_d I(w_d)$. That is, $q_{a,b}$ is a linear combination with positive coefficients of no more than $2(i+1)$ members from S.

We have completed the proof of Lemma 7. □

Define a function $F(i,j) = \lfloor\frac{1}{8}D(i,j)\rfloor$ for all $i,j \ge 0$. It follows from (9) that, for all $m \ge 4n$,

$$f(m,n) \le 2+2\min\{i \mid F(i,\lfloor m/(4n)\rfloor-1)>n\}. \quad (13)$$

Lemma 8. $F(i,j) \ge A(i,j)$ for all $i,j \ge 0$.

Proof. From Lemma 6 and 7, we obtain

$$F(0,j) \ge 2^{4j} \quad \text{for } j \ge 0,$$
$$F(i,0) \ge 1, \quad F(i,1) \ge 16 \quad \text{for } i \ge 1,$$

and

$$F(i,j) \ge 8F(i,j-1)F(i-1,F(i,j-1)) \text{ for } i\ge 1,\ j\ge 2.$$

As $F(i,j) = \lfloor\frac{1}{8}D(i,j)\rfloor \ge 1$, the last inequality implies

$$F(i,j) \ge F(i-1,F(i,j-1)) \text{ for } i\ge 1,\ j\ge 2.$$

One can then prove the lemma easily by induction on the lexicographical order of (i,j), making use of Lemma 1. □

We now use (13), Lemma 8, the fact $m \ge 20n$, and Lemma 2 to obtain

$$f(m,n) \le 2+2\ \min\{i \mid A(i,\lfloor m/(4n)\rfloor - 1 > n\}$$
$$\le 2+2a(n,\lfloor m/(4n)\rfloor - 1)$$
$$= 2+O(a(n,4\lfloor m/n\rfloor)).$$

To complete the proof of (8), we use Lemma 3 to obtain

$$f(m,n) = 2 + O(a(\log_2 n, 4\lfloor m/n\rfloor))$$
$$= 2 + O(\alpha(m,n))$$
$$= O(\alpha(m,n)).$$

B. The Case $m < 20n$.

We can assume that $m \ge n+20$, otherwise the bound $O(n/(m-n+1))$ is trivially an upper bound. Define r by $m = n + \frac{n}{r}$, and let $\ell = \lfloor \frac{n}{20r} \rfloor \ge 1$. Divide $(x_1,x_2,\ldots,x_n)$ into ℓ consecutive blocks $T_1,T_2,\ldots,T_\ell$ of almost equal size (differing by at most 1). Then $|T_i| = \lceil n/\ell \rceil$ or $\lfloor n/\ell \rfloor$, and hence

$$|T_i| \le 40r + 1 \quad \text{for all } i.$$

We now construct scheme $S = S_1 \cup S_2$, where $S_1 = \{z_i = x_i \mid 1 \le i \le n\}$ and S_2 will be defined below. Let τ be a $(t,m-n)$-scheme for ℓ items $(y_1,y_2,\ldots,y_\ell)$, where $t = f(m-n,\ell)$. For any linear combination w of y's, let $I(w)$ denote the resulted expression if we replace eacy y_i by $\sum_{x_s \in T_i} x_s$ in w. Define $S_2 = \{I(w) \mid w \in \tau\}$. The scheme S clearly uses space $|S_1| + |S_2| = m$. Using an argument similar to the proof of Lemma 7, one can easily verify that S has a retrieval time $f(m-n,\ell) + 2(40r)$. Using the result $f(20\ell,\ell) = O(\alpha(20\ell,\ell))$ established in Section 5.2A, we obtain

$$f(m,n) = O(f(20\ell,\ell) + \frac{n}{m-n+1})$$
$$= O(\alpha(20\ell,\ell) + \frac{n}{m-n+1}).$$

We complete the proof by observing that $n \ge \ell$ and hence

$$\alpha(20\ell,\ell) = a(\log_2 \ell, 80)$$
$$= O(a(\log_2 n, 80))$$
$$= O(\alpha(20n,n))$$
$$= O(\alpha(m,n)).$$

5.3 Lower Bound.

In this section we will show that

$$f(m,n) = \Omega(\alpha(m,n) + \frac{n}{m-n+1}) \text{ for } m \ge n \ge 1. \qquad (14)$$

Instead of the original formulation, let us consider a slightly different problem:

Let $X = (x_1,x_2,\ldots,x_n)$ be an n-tuple of distinct objects. A structure for X is a family of subsets $S = \{Z_1,Z_2,\ldots,Z_{m'}\}$, where $Z_i \subseteq \{x_1,x_2,\ldots,x_n\}$; we call S a (t,m)-structure for X if $m' \le m$ and every subset of the form $\{x_k \mid a \le i \le b\}$ is the union of t or less of the Z's. We will refer to t as the retrieval time and m as the space used by S. Given m,n, let $g(m,n)$ denote the minimum t for which a (t,m)-structure for X exists.

It is clear that $f(m,n) \ge g(m,n)$, because any (t,m)-scheme for X can be turned into a (t,m)-structure by identifying $\sum_i \lambda_i x_i$ with $\{x_i \mid \lambda_i \ne 0\}$. We devote the rest of the section to proving

$$g(m,n) = \Omega(\alpha(m,n) + \frac{n}{m-n+1}) \text{ for } m \ge n \ge 1, \qquad (15)$$

which implies (14).

A. Proof of $g(m,n) = \Omega(\alpha(m,n))$.

For any integers $i,j \ge 0$, let $G(i,j)$ be the largest integer $n \ge 0$ for which there exists an $(i+1,\lfloor jn/3 \rfloor)$-structure for $X = (x_1,x_2,\ldots,x_n)$. We will show that $G(i,j)$ are finite for all i,j and find an upper bound on the growth rate of G. We can then use the following obvious inequality to derive the desired lower bound:

$$g(m,n) \ge \min\{i+1 \mid G(i,\lfloor 3m/n \rfloor) \ge n\}. \qquad (16)$$

Lemma 9.

$$G(0,0) = 0,\quad G(0,j) \le j-1 \quad \text{for } j \le 1,$$
$$G(i,0) = G(i,1) = 0 \quad \text{for } i \le 1.$$

Proof. The only nontrivial statement is that $G(0,j) \le j-1$. This follows from the fact that, to achieve $t = 1$ for n items, a space of $\binom{n+1}{2}$ is needed. □

The crucial idea for the proof of the lower bound is contained in the next lemma.

Lemma 10. For each $i,j \ge 1$, $G(i,j)$ is finite and

$$G(i,j) \le 2(1+G(i,j-1))(1+G(i-1,12(1+G(i,j-1)))).$$

Proof. We will prove the lemma by induction on the lexicographical order of (i,j). Let $i \ge 1$, $j \ge 1$, and suppose the inequality is true for all (i',j') lexicographically less than (i,j); we will prove it for (i,j).

We can assume $j > 1$; otherwise it is trivially true by Lemma 9. Let $n > 2(1+G(i,j-1))(1+G(i-1,12(1+G(i,j-1))))$. Suppose there exists an $(i+1,\lfloor jn/3 \rfloor)$-structure $S = \{Z_1,Z_2,\ldots,Z_m\}$ for $X = (x_1,x_2,\ldots,x_n)$, where $m \le \lfloor jn/3 \rfloor$. We will derive a contradiction. That will complete the inductive step.

Denote $\{x_i | a \le i \le b\}$ by $\langle a,b\rangle$ for integers $1 \le a \le b \le n$. Without loss of generality, we can assume that each Z_i is of the form $\langle a,b\rangle$. (Otherwise, we can replace each Z_j by the smallest $\langle a,b\rangle$ containing Z_j, and obtain an $(i+1,\lfloor jn/3\rfloor)$-structure with this property.)

Let $\ell = \lfloor n/(1+G(i,j-1))\rfloor$, then $\ell > 1$. Divide the list $(x_1,x_2,\ldots,x_n)$ into ℓ consecutive blocks $T_1,T_2,\ldots,T_\ell$ of almost equal size (differing by at most 1). Then

$$|T_s| > G(i,j-1) \quad \text{for all } s. \tag{17}$$

The following elementary fact will be used later.

Fact A. If $V \subseteq \{1,2,\ldots,\ell\}$ and $|V| > \lceil \ell/2 \rceil$, then $\sum_{s\in V}|T_s| > n/3$.

Proof. All T_s are non-empty and $|T_s|/|T_{s'}| \le 2$ for all s, s'. □

Partition S into two parts S_1 and S_2, where

$$S_2 = \{\langle a,b\rangle | \langle a,b\rangle \in S,\ x_a \in T_r,\ x_b \in T_{r'} \text{ with } r<r'\},$$

with $S_1 = S - S_2$. Let us further partition S_1 into $S_{11} \cup S_{12} \cup \ldots \cup S_{1\ell}$, where

$$S_{1r} = \{Z_s | Z_s = \langle a,b\rangle \text{ for some } x_a, x_b \in T_r\}.$$

Fact B. S_{ir} is a structure with retrieval time $i+1$ for the block T_r.

Proof. Obvious. □

Fact C. $|S_2| \le n/3$.

Proof. Suppose otherwise, then $|S_1| < m - \frac{n}{3} \le (j-1)n/3$. It follows that there is some r with $|S_{1r}| < (j-1)|T_r|/3$. But (17) implies that S_{1r} cannot be an $(i+1,\lfloor (j-1)|T_r|/3\rfloor)$-structure. This contradicts Fact B. □

Consider now the blocks T_s as items, and let $T = (T_1,T_2,\ldots,T_\ell)$. Write $\langle\langle r,r'\rangle\rangle$ for the set $\{T_r,T_{r+1},\ldots,T_{r'}\}$. For any $Z = \langle a,b\rangle$ where $a \le b$, we define $\overline{Z} = \langle\langle r,r'\rangle\rangle$ when $x_a \in T_r$ and $x_b \in T_{r'}$. Define a structure A for T as follows.

$$A = \{\langle\langle r,r\rangle\rangle | 1 \le r \le \ell\} \cup \{\langle\langle r,r'\rangle\rangle | \langle\langle r,r'\rangle\rangle = \overline{Z} \text{ for some } Z \in S_2\}.$$

Fact D. A is an $(i+1,\ell+|S_2|)$-structure for T.

Proof. Immediate. □

By Fact D, every $\langle\langle r,r'\rangle\rangle$ can be written as the union of $t \le i+1$ members of A; let $d_{r,r'}$ denote the smallest such t. Call $\langle\langle r,r'\rangle\rangle$ costly if $d_{r,r'} = i+1$. We say a T_r is troublesome if there exists a $r' > r$ with costly $\langle\langle r,r'\rangle\rangle$.

Fact E. For each troublesome T_r and $x_a \in T_r$, there exists some $Z_s = \langle a,a_1\rangle \in S_2$.

Proof. Consider a costly $\langle\langle r,r'\rangle\rangle$, and choose any $x_b \in T_{r'}$. By the definition of S, one can write

$$\langle a,b\rangle = \bigcup_{s\in V} Z_s, \tag{18}$$

with $|V| \le i+1$. Clearly, there exists a Z_s (with $s \in V$) of the form $\langle a,a_1\rangle$. We claim that $Z_s \in S_2$ for all $s \in V$. Otherwise, let $V' = \{s | Z_s \in S_2,\ s \in V\}$, then $|V'| < i+1$. Formula (18) then implies

$$\langle\langle r,r'\rangle\rangle = \bigcup_{s\in V'} \overline{Z}_s.$$

But this contradicts the fact that $\langle\langle r,r'\rangle\rangle$ is costly. This proves Fact E. □

Fact F. There are at most $\lfloor \ell/2\rfloor$ troublesome T_r.

Proof. Suppose this is false. Let $L = \{a | x_a \in T_r$ for troublesome $T_r\}$. By Fact A, we have

$$|L| > \frac{1}{3}n.$$

By Fact E, we have

$$|S_2| \ge |L| > \frac{1}{3}n.$$

This contradicts Fact C. □

For any subset $W \subseteq T$, let W' denote the resulted subset after all the troublesome T_r have been deleted from W. Consider T' and the structure for T' defined by $\{R' | R \in A\}$; call this structure A'. By Fact F, $|T'| \ge \ell - \lfloor \ell/2\rfloor = \lceil \ell/2\rceil$. Clearly, A' is an $(i,|A|)$-structure for T'. By Fact D, $|A| \le \ell + |S_2| \le \ell + \frac{n}{3}$. Noting that $\ell = \left\lfloor \frac{n}{1+G(i,j-1)} \right\rfloor > 1$, we have

$$\frac{\ell+\frac{n}{3}}{\ell} = 1 + \frac{\frac{n}{3}}{\left\lfloor \frac{n}{1+G(i,j-1)} \right\rfloor} \le 2(1+G(i,j-1)).$$

As $|T'| \ge \lceil \ell/2\rceil$, we have

$$\begin{aligned} |A| &\le \ell + \frac{n}{3} \\ &\le 2(1+G(i,j))\ell \\ &\le 4(1+G(i,j-1))|T'|. \end{aligned}$$

This means A' is an $(i,4(1+G(i,j-1))|T'|)$-structure for T'. However, we have clearly $|T'| \ge \ell/2 > G(i-1,12(1+G(i,j-1)))$. This is a contradiction to the definition of G.

We have completed the induction step in the proof of Lemma 10. □

We now use the information contained in Lemmas 9 and 10 to give a bound on $G(i,j)$. Recall the function $H(i,j)$ defined in Section 5.1.

Lemma 11. $H(i,j) \geq G(i,j)$ for $i \geq 0$, $j \geq 1$.

Proof. Let $L(i,j) = 1+G(i,j)$. We will prove by induction the following stronger statement: For $i \geq 0$, $j \geq 1$,

$$H(i,j) \geq 12j^2L(i,j). \qquad (19)$$

Inequality (19) is clearly true for $i = 0$. Let $i > 0$, $j \geq 1$, and suppose it is true for all (i',j') lexicographically less than (i,j). We will prove that it is true for (i,j).

The case $j = 1$ is easily verified. We thus assume $j > 1$. Then, using the inductive hypothesis, the fact L and H are monotone, and Lemmas 9 and 10, we obtain

$$\begin{aligned}
H(i,j) &= H(i-1,H(i,j-1)) \\
&\geq 12(H(i,j-1))^2L(i-1,H(i,j-1)) \\
&\geq 12(H(i,j-1))^2L(i-1,12(j-1)^2L(i,j-1)) \\
&\geq 12H(i,j-1)\cdot 12(j-1)^2L(i,j-1) \\
&\qquad L(i-1,12L(i,j-1)) \\
&\geq 144H(1,1)(j-1)^2L(i,j-1) \\
&\qquad L(i-1,12L(i,j-1)) \\
&\geq 12j^2\cdot 2(1+G(i,j-1))(1+G(i-1,12L(i,j-1))) \\
&\qquad + 12j^2 \\
&\geq 12j^2G(i,j)+12j^2 \\
&= 12j^2L(i,j).
\end{aligned}$$

It follows from (16), Lemma 11, and Lemma 5 that

$$\begin{aligned}
g(m,n) &\geq \min\{i+1 \mid H(i,\lceil 3m/n\rceil) \geq n\} \\
&\geq \min\{i+1 \mid A(i+4,\lceil 3m/n\rceil + 4) > n\} \\
&= a(n,\lceil 3m/n\rceil + 4) - 3.
\end{aligned}$$

Using Lemma 2, we obtain

$$\begin{aligned}
g(m,n) &= \Omega(a(n,4\lceil m/n\rceil)) - 3 \\
&= \Omega(a(\log_2 n, 4\lceil m/n\rceil)) - 3 \\
&= \Omega(\alpha(m,n)) - 3.
\end{aligned}$$

As $g(m,n) \geq 1$, we have proved

$$g(m,n) = \Omega(\alpha(m,n)).$$

B. Proof of $g(m,n)=\Omega(n/(m-n+1))$.

We can assume that $m - n \leq \frac{1}{10}n$ and $n > 10$; otherwise this lower bound is trivial. Write $m = n+p$. Consider any structure $S = \{Z_1,Z_2,\ldots,Z_m\}$ for $X = (x_1,x_2,\ldots,x_n)$. We will exhibit an $\langle a,b\rangle$ such that $\langle a,b\rangle = \cup_{s\in V}Z_s$ implies $|V| \geq \frac{n-1}{m-n+1} - 1$. This would prove the desired lower bound.

Without loss of generality, we assume that

$$Z_i = \langle a_i,b_i\rangle, \qquad 1 \leq i \leq p,$$

$$Z_{p+j} = \{x_j\}, \qquad 1 \leq j \leq n,$$

where $1 \leq b_1 \leq b_2 \leq \ldots \leq b_p \leq n$. Define $b_0 = 1$ $b_{p+1} = n$. Consider the gaps b_1-b_0, $b_2-b_1,\ldots,$ $b_{p+1}-b_p$; let $b_{j+1} - b_j$ be the largest gap. Then

$$b_{j+1}-b_j \geq \frac{n-1}{p+1} = \frac{n-1}{m-n+1}.$$

Clearly $b_{j+1} \geq 4$. Consider the set $\langle 1,b_{j+1}-1\rangle$. If

$$\langle 1,b_{j+1}-1\rangle = \cup_{s\in V} Z_s,$$

then $\{p+k \mid b_j < k \leq b_{j+1}\} \subseteq V$. This means

$$|V| \geq b_{j+1}-b_j-1 \geq \frac{n-1}{m-n+1} - 1.$$

This proves the lower bound of $\Omega(n/(m-n+1))$.

We have completed the proof of (15).

6. PROOF OF THEOREM 2.

The proof of Theorem 1 can easily be modified to prove Theorem 2. We first discuss the upper bound. In the proof we have implicitly given a recursive construction of a (t,m)-scheme S for n items, with $t = O(\alpha(m,n)+\frac{n}{m-n+1})$. It is easy to verify inductively that the scheme $S = \{z_1,z_2,\ldots\}$ has the following properties: (1) For each i, $z_i = \sum_{j\in V_i} x_j$ for some $V_i \subseteq \{1,2,\ldots,n\}$; (2) Every $q_{ij} = x_i+x_{i+1}+\ldots+x_j$ can be written as $z_{k_1} + z_{k_2} + \ldots + z_{k_s}$ with $s \leq t$ mutually disjoint $V_{k_1},V_{k_2},\ldots,V_{k_s}$. Clearly, such a (t,m)-scheme will also serve as a (t,m)-scheme for n items of any semigroup S. This proves the upper bound $f_S(m,n) = O(\alpha(m,n) + \frac{n}{m-n+1})$.

To establish the lower bound for $f_S(m,n)$, we will argue that, from any (t,m)-scheme S for n items $(x_1,x_2,\ldots,x_n)$ of a faithful semigroup S, we can obtain a (t,m)-structure τ on $(x_1,x_2,\ldots,x_n)$. This will prove $f_S(m,n) = \Omega(\alpha(m,n) + \frac{n}{m-n+1})$ immediately.

For any linear expression $y = \sum_j n_jx_j$ with integral $n_j \geq 0$, let $Z(y)$ denote the set $\{x_j \mid n_j > 0\}$. Given any (t,m)-scheme S on $(x_1,x_2,\ldots,x_n)$, we define $F = \{Z(y) \mid y \in S\}$. Then F is a structure for $(x_1,x_2,\ldots,x_n)$ using space m in the sense defined in Section 4. Observe that if

$$x_i+x_{i+1}+\ldots+x_j = \sum_{y\in S_1\subseteq S} \lambda_y y$$

is an identity for $x_1,x_2,\ldots,x_n \in S$ with integral $\lambda_y > 0$, then we can use the faithfulness of S to conclude

$$\{x_i,x_{i+1},\ldots,x_j\} = \bigcup_{y\in S_1} Z(y).$$

This fact implies that τ is a (t,m)-structure. We have thus completed the lower bound proof.

7. CONCLUDING REMARKS

The study of data structure is an important aspect of designing computer algorithms. Over the years, many ingenious data structures have been invented for various purposes. On the other hand, the complexity of data manipulation has just begun to be explored. Fredman's model [6] for range query presents a unified framework for a fairly broad set of problems. In the present paper, we have begun to examine the intrinsic space-time tradeoff question for range queries in this general framework, and have used it to prove several new results on dynamic queries. There are many open questions in this direction waiting to be solved. We will only mention a few:

(A) What is the precise space-time tradeoff for circular query? The performance of the known data structure comes nowhere near the lower bound we derived [2].

(B) What is the space-time tradeoff for orthogonal queries? Can we prove that if $S = O(n)$, then $T = \Omega((\log n)^{d-1})$?

(C) What can be said when G is a group instead of a semigroup (i.e., with subtraction allowed)? Some results in this direction can be found in [4], but we know much less about range queries when the data belong to a group. In particular, does Theorem 3 remain valid in the group model?

REFERENCES

[1] J.L. Bentley and H.A. Mauer, "Efficient worst-case data structure for range searching," Acta Informatica 13 (1980), 155-168.

[2] J.L. Bentley and H.A. Mauer, "A note on Euclidean near neighbor searching in the plane," Information Processing Letters 8 (1979), 133-136.

[3] M.L. Fredman, "Lower bounds on the complexity of some optimal data structures," SIAM J. on Computing 10 (1981), 1-10.

[4] M.L. Fredman, "The complexity of maintaining an array and computing its partial sums," to appear in Journal of ACM.

[5] M.L. Fredman, "Query time versus redundancy trade-offs for range queries," to appear in Journal of Computer and System Sciences, 1981.

[6] M.L. Fredman, "The inherent complexity of dynamic data structures which accomodata range queries," Proc. 21st Annual IEEE Symposium on Foundations of Computer Science (1980), 191-200.

[7] G.S. Leuker, "A data structure for orthogonal range queries," Proc. 19th Annual IEEE Symposium on Foundation of Computer Science (1978), 28-34.

[8] R.L. Rivest, "Partial match retrieval algorithms," SIAM J. on Computing 5(1976), 19-50.

[9] R.E.Tarjan, "Efficiency of a good but not linear set union algorithm," Journal of ACM 22 (1975), 215-225.

[10] R.E. Tarjan, "Complexity of monotone networks for computing conjunctions," Algorithmic Aspects of Combinatorics edited by B. Alspach, P. Hell, and D.J. Miller, North-Holland (1978), 121-134.

[11] D.E. Willard, "Polygon retrieval," to appear in SIAM J. on Computing.

[12] D.E. Willard, "New data structures for orthogonal range queries," Tech Report TR-22-78, (1978), Aiken Computation Lab., Harvard University.

[13] A.C. Yao, "On the complexity of maintaining partial sums," IBM San Jose Research Center Report RJ3228, September 1981.

[14] A.C. Yao, "On the complexity of answering circular range queries," to appear as an IBM report.