

On Searching Sorted Lists: A Near-Optimal Lower Bound

Paul Beame*
Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA
USA 98195-2350

Faith Fich†
Computer Science Department
University of Toronto
Toronto, Ontario
CANADA M5S 1A4

April 23, 1997

Abstract

We obtain improved lower bounds for a class of static and dynamic data structure problems that includes several problems of searching sorted lists as special cases.

These lower bounds nearly match the upper bounds given by recent striking improvements in searching algorithms given by Fredman and Willard's fusion trees [9] and Andersson's search data structure [5]. Thus they show sharp limitations on the running time improvements obtainable using the unit-cost word-level RAM operations that those algorithms employ.

1 Introduction

Traditional analysis of problems such as sorting and searching is often schizophrenic in dealing with the operations one is permitted to perform on the input data. In one view, the elements being sorted are seen as abstract objects which may only be compared. In the other view, one is able to perform certain word-level operations, such as indirect addressing using the elements themselves, in algorithms like bucket and radix sorting.

Traditionally, the second view is only applied when the number of bits to represent the data is very small in comparison with the number of elements being sorted. More recently, algorithms such as the fusion tree sorting algorithm of Fredman and Willard [9] and subsequent algorithms in [10, 7, 3], have shown that one can obtain significant speed-ups by fully exploiting unit-cost word-level operations and the fact that data elements being sorted need to fit in words of memory. For example, the fusion tree data structure of Fredman and Willard can store a list of s elements in $O(s)$ words of memory in such a way that range searching can be accomplished in amortized sub-logarithmic ($O(\sqrt{\log s})$) time using word-level operations. This data structure forms the basis of their ingenious $O(n\sqrt{\log n})$ sorting algorithm. Fusion Trees require word-level multiplication as a

*Research supported by the National Science Foundation under grant CCR-9303017

†Research supported in part by grants from the Natural Sciences and Engineering Research Council and the Information Technology Research Centre of Ontario

unit-cost operation, which may be more than one wishes to allow, but Andersson [5], building on an $O(n \log \log n)$ time sorting algorithm in this model in [3], has shown that even a simpler collection of word-level operations (all of them computable in AC^0) permits the $O(\sqrt{\log s})$ running time and $O(s)$ space as a worst case bound. Andersson’s algorithm is not particularly complicated; in fact, his paper even includes most of the C code that implements one variant of his sub-logarithmic range searching!

Because these new algorithms fully exploit the freedom to use word-level operations, standard lower bounds for sorting and searching do not always apply to them. What are the limits to the improvements in running-time obtained using this freedom?

One of the most natural and general models for proving lower bounds for data structures problems, and one that is ideally suited for representing word-level operations, is the *cell-probe* model, introduced by Yao [18]. In this model, there is a memory consisting of *cells*, each of which is capable of storing some fixed number of bits. A cell-probe algorithm is a decision tree with one memory cell accessed at each node. The decision tree branches according to the contents of the cell accessed. We only count the number of memory cells accessed in the data structure; all computation is free. This means that no restrictions are imposed on the way data is represented or manipulated, except for the bound on the size of values that each memory cell can hold. Thus, lower bounds obtained in this model apply to all reasonable models of computation and give us insight into why certain problems are hard.

Both static problems (just queries) and dynamic problems (updates and queries) have been examined in the cell-probe model. (For static problems, it is also necessary to impose a bound on the number of memory cells in order for the model to be meaningful; otherwise a precomputed table of answers can be used to solve any given problem with only one access.) A number of authors have considered lower bounds in the cell-probe model for both static and dynamic data structure problems [18, 2, 8, 13].

The particular class of problems that we consider includes problems such as both the static and dynamic PREDECESSOR and ONE-DIMENSIONAL RANGE QUERY problems. For example, the PREDECESSOR problem, the classical problem solved by binary search on sorted lists, requires one to maintain a set S from a universe U in such a way that queries of the form “Is universe element j an element of S and, if not, what element of S , if any, is just before it in sorted order?” may be answered efficiently.

Van Emde Boas et al. [14] showed that the dynamic version of the PREDECESSOR problem (as well as the more general priority queue problem) in a universe of size n may be solved on a RAM at a cost of only $O(\log \log n)$ time per update. Willard [15] extended this result to show that if the set S has size at most s then, in the static version of the PREDECESSOR problem, only $O(s)$ memory cells are needed so that each query can be answered in $O(\log \log n)$ time on a RAM. Ajtai, Fredman, and Komlos [1] also considered the static problem. They showed that, if n , the size of the universe, is only polynomial in $s = |S|$ then, using a trie, one can store S in $O(s)$ memory cells and answer PREDECESSOR queries in constant time in the cell-probe model.

In [2], on which many of the techniques of this paper are based, Ajtai showed that, even in the very general cell-probe model, the result of Ajtai, Fredman, and Szemerédi cannot be extended to general S , even if the memory size is permitted to be as large as $s^{O(1)}$. More precisely, he showed that if the size n of the universe U grows more than polynomially in $s = |S|$, then any cell-probe

data structure using $s^{O(1)}$ memory cells requires non-constant time per query.

In [13], Miltersen defined a natural class of data structure problems that includes the PREDECESSOR and ONE-DIMENSIONAL RANGE QUERY problems as special cases and he generalized many of the results above to this class of problems. He showed that Willard’s translation of dynamic data structures to static data structures can be extended to derive lower bounds on dynamic data structure problems in this class. As well, he observed that, by choosing parameters as in Ajtai’s argument, one can find a set size s , in terms of the universe size n , for which the time per query derived from Ajtai’s argument is $\Omega(\sqrt{\log \log n})$. This lower bound on query time also applies to the dynamic case when update time is at most $2^{(\log n)^{1-\epsilon}}$, for some constant $\epsilon > 0$.

One of the other nice contributions of Miltersen’s paper is the observation that cell-probe algorithms can be viewed as two-party communication protocols [17] between a Querier who holds the input query and a Responder who holds the data structure. In each round of communication, the Querier sends the name of a memory cell to access and the Responder answers with the contents of that memory cell. (The only restriction is that the same query always receives the same response.) He rephrases most of Ajtai’s argument in this form where its similarities with the communication-complexity lower bound argument of Karchmer and Wigderson [11] become quite apparent.

Fusion trees and Andersson’s search structure also provide solutions to the static PREDECESSOR problem. They use $O(\sqrt{\log s})$ time per query. Unlike the PREDECESSOR data structure in [1], these structures use only $O(s)$ memory cells for a given set size s , *independent of universe size* (provided that elements of the universe fit in individual words of memory.)

Miltersen et al. [12] observed that, for certain universe sizes, the lower bound of [13] also gives an $\Omega((\log s)^{1/3})$ lower bound on the time to implement any static problem from the class considered in [13] and thus on search data structures, such as fusion trees, with an $s^{O(1)}$ bound on the number of memory cells.

In this paper, we nearly close the gaps between the upper and lower bounds for these data structure problems in the cell-probe model. We improve the lower bound in terms of universe size to $\Omega(\log \log n / \log \log \log n)$ which nearly matches the $O(\log \log n)$ upper bound on RAMs of Van Emde Boas et al. In terms of the number of elements, s , we improve the bound to $\Omega(\sqrt{\log s / \log \log s})$ nearly matching the $O(\sqrt{\log s})$ upper bound on augmented RAMs due to Andersson.

The basic structure of our argument follows that of [2] as expressed by [13]. Our key improvement is that we find a better distribution of inputs on which to consider the data structure’s behaviour.

2 Preliminaries

In this section, we state two combinatorial results which are important for the lower bound proofs given in the next section. We use the notation $[1, n]$ to denote the set of integers $\{1, \dots, n\}$ and $(a, a']$, for $a < a'$, to denote the set of integers $\{a + 1, \dots, a'\}$.

The following form of the Chernoff-Hoeffding bound follows easily from the presentation in [6].

Proposition 1: Fix $H \subseteq U$ with $|H| \geq \rho|U|$ and let $S \subseteq U$ with $|S| = s$ be chosen uniformly at

random. Then

$$\Pr[|H \cap S| \leq \rho s/4] \leq (\sqrt{2}/e^{3/4})^{\rho s} < 2^{-\rho s/2}.$$

The next result is a small modification and rephrasing of a combinatorial lemma that formed the basis of Ajtai's lower bound argument in [2].

Suppose we have a tree T of depth d such that all nodes on the same level have the same number of children. For $\ell = 0, \dots, d$ let V_ℓ be the set of nodes of T on level ℓ (i.e. at depth ℓ) and for $\ell < d$ let f_ℓ be the fan-out of each node on level ℓ . Thus $|V_{\ell+1}| = f_\ell |V_\ell|$ for $\ell = 0, \dots, d-1$.

For any node $v \in T$, let $\text{leaves}(v)$ denote the set of nodes in V_d that are descendants of v and, for v not the root of T , let $\text{parent}(v)$ denote the parent of v . Let $A(1), \dots, A(m)$ be disjoint sets of leaves of T and let $A = \bigcup_{c=1}^m A(c)$. The leaves in $A(c)$ are said to have *colour* c . A nonleaf node v has *colour* c if $\text{leaves}(v)$ contains a node in $A(c)$. For $c = 1, \dots, m$, let $A'(c) = \{v \mid \text{leaves}(v) \cap A(c) \neq \emptyset\}$ denote the set of nodes with colour c . Note that the sets $A'(1), \dots, A'(m)$ are not necessarily disjoint, since a nonleaf node may have more than one colour.

A nonleaf node v is δ -dense (where $0 \leq \delta \leq 1$) if there is a colour c such that at least a fraction δ of v 's children have colour c , i.e., if v is on level ℓ then v has at least $\delta \cdot f_\ell$ children in $A'(c)$.

Let $R_\ell^\delta(c)$ be the set of those nodes on level ℓ that are coloured c and do not have a δ -dense ancestor at levels $1, \dots, \ell-1$. In particular, $R_1^\delta(c) = A'(c) \cap V_1$. The fraction of nodes on level ℓ that are in $R_\ell^\delta(c)$ decreases exponentially with ℓ .

Proposition 2: For $1 \leq \ell \leq d$, $|R_\ell^\delta(c)| \leq \delta^{\ell-1} |V_\ell|$.

Proof By induction on ℓ . The base case, $\ell = 1$, is trivial since $R_1^\delta(c) \subseteq V_1$.

Now let $1 \leq \ell < d$ and assume that $|R_\ell^\delta(c)| \leq \delta^{\ell-1} |V_\ell|$. If $v \in R_{\ell+1}^\delta(c)$, then, by definition, v has colour c and no ancestor of v at levels $1, \dots, \ell$ is δ -dense. Since v has colour c , $\text{parent}(v)$ also has colour c and, thus, $\text{parent}(v) \in R_\ell^\delta(c)$. Furthermore, $\text{parent}(v)$ is not δ -dense, so fewer than $\delta \cdot f_\ell$ of its children are in $R_{\ell+1}^\delta(c)$. Hence,

$$|R_{\ell+1}^\delta(c)| < \delta \cdot f_\ell |R_\ell^\delta(c)| \leq \delta \cdot f_\ell \cdot \delta^{\ell-1} |V_\ell| = \delta^\ell |V_{\ell+1}|,$$

as required. \square

We now prove Ajtai's Lemma:

Proposition 3: (Ajtai's Lemma) Let T be a tree of depth $d \geq 2$ such that all nodes on the same level of T have the same number of children. Suppose that at least a fraction α of all the leaves in T are coloured (each with one of m colours). Call this set of leaves A . Then there exists a level ℓ , $1 \leq \ell \leq d-1$, such that the fraction of nodes on level ℓ of T that are δ -dense is at least

$$\frac{\alpha - m\delta^{d-1}}{d-1}.$$

Proof By Proposition 2, $|R_d^\delta(c)| \leq \delta^{d-1} |V_d|$ for all colours c . Let $R^\delta = \bigcup_{c=1}^m R_d^\delta(c) \subseteq A$. There are m colours; therefore $|R^\delta| \leq m\delta^{d-1} |V_d|$.

If $w \in A \subseteq V_d$ and none of its ancestors at levels $1, \dots, d-1$ are δ -dense, then $w \in R^\delta$. Thus $w \in A - R^\delta$ implies that some ancestor of w at some level $1, \dots, d-1$ is δ -dense.

For $\ell = 1, \dots, d-1$, let δ_ℓ denote the fraction of nodes in V_ℓ that are δ -dense. Observe that because the fan-out at each level of T is constant, for any $v \in V_\ell$, $|\text{leaves}(v)| = |V_d|/|V_\ell|$. Therefore, for each ℓ , $1 \leq \ell \leq d-1$, the number of leaf nodes of T that lie below δ -dense nodes in V_ℓ is $\delta_\ell |V_d|$. It follows that

$$|A - R^\delta| \leq \sum_{\ell=1}^{d-1} \delta_\ell |V_d|.$$

But $|A - R^\delta| = |A| - |R^\delta| \geq \alpha |V_d| - m\delta^{d-1}|V_d|$, so

$$\sum_{\ell=1}^{d-1} \delta_\ell \geq \alpha - m\delta^{d-1}.$$

Thus there is some ℓ , $1 \leq \ell \leq d-1$, such that $\delta_\ell \geq (\alpha - m\delta^{d-1})/(d-1)$, as required. \square

3 Lower Bounds for Static Problems

As in Miltersen [13], we prove our lower bounds in a general language-theoretic setting. At the end of the section, we show how to obtain the results for the specific data structure problems mentioned in the introduction as corollaries of these lower bounds.

Throughout, we assume that Σ is a finite alphabet that does not contain the symbol \perp .

DEFINITION 3.1: A regular language L is *indecisive* if and only if for all $x \in \Sigma^*$ there exist $z, z' \in \Sigma^*$ such that $xz \in L$ and $xz' \notin L$. In other words, knowing the prefix of any word does not determine whether the word is in L .

If the minimal deterministic finite automaton accepting L has q states, then the strings z and z' may be taken to be of length at most $q-1$ in Definition 3.1. Furthermore, $q \geq 2$, since the deterministic finite automaton must accept some words and reject others.

DEFINITION 3.2: For any string $y = y_1 \cdots y_n \in (\Sigma \cup \{\perp\})^*$ and any nonnegative integer $j \leq n$, we use the notation $\text{PRE}_j(y)$ to denote the string obtained by deleting all occurrences of \perp from $y_1 \cdots y_j$, the length j prefix of y .

DEFINITION 3.3: Let $Z(n, s)$ denote the set of strings in $(\Sigma \cup \{\perp\})^n$ containing at most s non- \perp characters.

DEFINITION 3.4: The *static (L, n, s) -prefix problem* is to store an arbitrary string $y \in Z(n, s)$ so that, for any $j \in [1, n]$, the query “Is $\text{PRE}_j(y) \in L$?” may be answered.

If $n' \geq n$ and $s' \geq s$, then any string in $Z(n, s)$ can be viewed as a string in $Z(n', s')$ by appending $\perp^{n'-n}$ to it. Thus the static (L, n', s') -prefix problem is at least as hard as the static (L, n, s) -prefix problem.

We prove a lower bounds on the complexity of the static (L, n, s) -prefix problem using an adversary argument. As discussed in the introduction, Miltersen [13] observed that one can phrase a static data structure algorithm in the cell-probe model in terms of a communication protocol between two players: the Querier, who holds the information about the input to be dealt with, and the Responder, who holds the data structure. Each query that the Querier makes to the data structure, a cell name, consists of $\log m$ bits of communication, where m is the number of memory cells, and each response by the Responder, the contents of that named cell, consists of exactly b bits of communication. We say that this cell-probe communication protocol uses m *memory cells of b bits*. For technical reasons, we require that, at the end of the protocol, both the Querier and the Responder know the answer to the problem instance. The number of rounds of alternation of the players is the time t of the cell-probe communication protocol. Since we are considering decision problems, the answer to each problem instance is a single bit. In this case, once one of the players knows the answer, the other player can be told the answer in the next round. Therefore, if there is an algorithm for the static (L, n, s) -prefix problem that uses at most $t - 1$ probes, then there is a cell-probe communication protocol that solves this problem in at most t rounds.

The lower bound, in the style of [11], works ‘top down’, maintaining, for each player, a relatively large set of inputs on which the communication is fixed. Unlike [11], we actually have a non-uniform distributions on the Responder’s inputs, so our notion of ‘large’ is with respect to these distributions. The distributions change (get simpler) as the rounds of the communication proceed.

We find the following notation convenient for this.

DEFINITION 3.5: If D is a probability distribution on a set Z and $B \subseteq Z$, define $\mu_D(B)$ to be the probability that D chooses an element of B .

The simplest probability distribution on the Responder’s input that we consider is $D(n, s)$, which is defined as follows:

DEFINITION 3.6: An element $y = y_1 \dots y_n$ from $Z(n, s)$ is chosen according to $D(n, s)$ by first choosing $S \subseteq [1, n]$ uniformly at random with $|S| = s$, then, independently for each $j \in S$, choosing $y_j \in \Sigma \cup \{\perp\}$ uniformly at random, and finally setting $y_j = \perp$ for $j \notin S$. That is, each of the $\binom{n}{i} |\Sigma|^i$ strings in $Z(n, s)$ with exactly i non- \perp characters has probability $\binom{n-i}{s-i} / [\binom{n}{s} (|\Sigma| + 1)^s]$.

First we argue that there does not exist a large set of positions $A \subseteq [1, n]$ and a large set of strings $B \subseteq Z(n, s)$ for which the static (L, n, s) -prefix problem can be solved, for all $j \in A$ and $y \in B$, without communication. We actually consider a slightly more general problem: determining whether $x \cdot \text{PRE}_j(y) \in L$.

Lemma 4: Suppose that $L \subseteq \Sigma^*$ is an indecisive regular language accepted by a deterministic finite automaton with q states. Let $n \geq s > 0$ and suppose that $b \geq 2(|\Sigma| + 1)^q$, $\alpha \geq \max(8qb^2/s, 12b^3/s)$, and $\beta \geq 2^{-2b+1}$. Consider any set of positions $A \subseteq [1, n]$, with $|A| \geq \alpha n$, and any set of strings $B \subseteq Z(n, s)$, with $\mu_{D(n,s)}(B) \geq \beta$. Then, for any $x \in \Sigma^*$, there exist integers $a, a' \in A$ and a string $y \in B$ such that $x \cdot \text{PRE}_a(y) \in L$ and $x \cdot \text{PRE}_{a'}(y) \notin L$.

Proof Let $x \in \Sigma^*$. Consider the event that a string y randomly chosen from the distribution $D(n, s)$ has $x \cdot \text{PRE}_a(y) \in L$ for all $a \in A$ or $x \cdot \text{PRE}_a(y) \notin L$ for all $a \in A$. We will show that the probability of this event is less than β . Since $\mu_{D(n,s)}(B) \geq \beta$, it will follow that there exist integers $a, a' \in A$ and a string $y \in B$ such that $x \cdot \text{PRE}_a(y) \in L$ and $x \cdot \text{PRE}_{a'}(y) \notin L$.

It is convenient to restrict attention to a well spaced subset of A . Specifically, since $|A| \geq \alpha n \geq \alpha s > 2b^2$, it is possible to choose $a_0, \dots, a_{b^2} \in A$ at least distance $\lfloor (\alpha n - 1)/b^2 \rfloor \geq \alpha n/b^2 - 1 > \alpha n/(2b^2)$ apart from one another. Say $a_0 < \dots < a_{b^2}$. Then $|(a_{i-1}, a_i]| = a_i - a_{i-1} > \alpha n/(2b^2)$ for $i = 1, \dots, b^2$.

Let $S \subseteq [1, n]$ with $|S| = s$ be chosen uniformly at random. Then, since $q \leq \alpha s/(8b^2)$ and $\alpha s/(4b^2) \geq 3b$, applying Proposition 1 with $H = (a_{i-1}, a_i]$ and $\rho = \alpha/(2b^2)$,

$$\text{Prob}[|(a_{i-1}, a_i] \cap S| < q] \leq \text{Prob}[|(a_{i-1}, a_i] \cap S| \leq \alpha s/(8b^2)] < 2^{-\alpha s/(4b^2)} \leq 2^{-3b}.$$

Since $b \geq 4$, there are $b^2 \leq 2^b$ intervals. Therefore, the probability that at least one of them contains fewer than q elements of S is less than $b^2 2^{-3b} \leq 2^{-2b} \leq \beta/2$.

Now consider any fixed choice for S that has at least q elements in each of these b^2 intervals. For each interval $(a_{i-1}, a_i]$, consider the set Q_i of the last q elements of S in the interval. L is an indecisive regular language. Therefore, for each fixed choice w for the symbols of y that occur before the first element of Q_i , there are strings $z, z' \in (\Sigma \cup \{\perp\})^q$ such that $x \cdot \text{PRE}_{a_i}(wz) \in L$ and $x \cdot \text{PRE}_{a_i}(wz') \notin L$. There are $(|\Sigma| + 1)^q$ equally likely ways that the characters of y in positions indexed by Q_i will be assigned values. Thus, with probability at least $(|\Sigma| + 1)^{-q}$, either $x \cdot \text{PRE}_{a_{i-1}}(y) \in L$ and $x \cdot \text{PRE}_{a_i}(y) \notin L$ or $x \cdot \text{PRE}_{a_{i-1}}(y) \notin L$ and $x \cdot \text{PRE}_{a_i}(y) \in L$. Therefore, the probability that this event does not occur is at most $1 - (|\Sigma| + 1)^{-q}$.

Since the choices of the portions of string y in each of the b^2 intervals $(a_{i-1}, a_i]$ are independent, the probability that either $x \cdot \text{PRE}_{a_i}(y) \in L$ for all $i = 0, \dots, b^2$ or $x \cdot \text{PRE}_{a_i}(y) \notin L$ for all $i = 0, \dots, b^2$ is at most

$$(1 - (|\Sigma| + 1)^{-q})^{b^2} \leq e^{-(|\Sigma|+1)^{-q}b^2} < 2^{-2b} \leq \beta/2 < \beta$$

since $b \geq 2(|\Sigma| + 1)^q$.

Because $\mu_{D(n,s)}(B) \geq \beta$, it follows that there exists a string $y \in B$ such that neither $x \cdot \text{PRE}_a(y) \in L$ for all $a \in A$ nor $x \cdot \text{PRE}_a(y) \notin L$ for all $a \in A$. \square

Let $L \subseteq \Sigma^*$ be an indecisive regular language. For any positive integers b, k, t, n , and s define

- $\alpha = s^{-1/(4t)}$
- $\beta = 2^{-2b+1}$
- $u = 8kt$
- $r = b^2 u / \alpha$
- $f = b^3 u^3 / \alpha^2$.
- $n_0 = n; s_0 = s$

- and for $i \leq t$ define $n_{i+1} = (n_i/f)^{1/u}$ and $s_{i+1} = s_i/(ru)$.

We say that the tuple of parameters (b, k, t, n, s) satisfies the *integrality condition* if $1/\alpha$ is an integer greater than 1 and, for every $i \leq t$, n_i and s_i are integers and $n_i \geq s_i$.

If s is the $4t$ -th power of an integer larger than 1, then $1/\alpha$ is an integer greater than 1 and f and r are also integers. Since $f \geq ru$ and $u \geq 1$, the condition $n_t \geq s_t$ implies that $n_i \geq s_i$ for all $i \leq t$. Furthermore, if n_t and s_t are both integers, then $s_i = (ru)^{t-i} s_t$ and $n_i = f^{(u^{t-i}-1)/(u-1)} n_t^{u^{t-i}}$ for $i = 0, \dots, t-1$ are all integers. In particular, the integrality condition will hold for (b, k, t, n, s) if s is the $4t$ -th power of an integer larger than 1 and there are integers $n_t \geq s_t$ such that $s = (ru)^t s_t$ and $n = f^{(u^t-1)/(u-1)} n_t^{u^t}$.

Suppose that the integrality condition holds for (b, k, t, n, s) . We will define a distribution on $Z(n, s)$ and use it to demonstrate that no cell-probe communication protocol using s^k memory cells of b bits can solve the static (L, n, s) -prefix problem in t rounds.

The general idea of the lower bound argument will be to find, after each round, a portion of the Querier's and Responder's inputs on which the cell-probe communication protocol has made little progress. After i rounds, the possible values of the Querier's input in this portion will correspond to an interval $[1, n_i]$ of a smaller prefix problem for language L . After i rounds, the Responder's input for this smaller prefix problem will have at most s_i elements of Σ and thus be a member of $Z(n_i, s_i)$.

We begin with a distribution D_0 on the domain of the Responder's inputs $Z(n_0, s_0)$. After i rounds of the cell-probe communication protocol, the distribution on the portion of the Responder's inputs that remains of interest, $Z(n_i, s_i)$, will be denoted by D_i . The final distribution, D_t , will be $D(n_t, s_t)$. After t rounds, we will apply Lemma 4 to the remaining portion, to complete the lower bound.

We define the probability distribution D_i on $Z(n_i, s_i)$ inductively, for $i = t-1, \dots, 0$. For every $i < t$, each string in $Z(n_i, s_i)$ will be thought of as labelling the leaves of a tree T_i with depth $u+1$, having fan-out f at the root and having a complete n_{i+1} -ary tree of depth u at each child of the root. We choose a random element of D_i as the sequence of leaf labels of the tree T_i , which we label using the distribution D_{i+1} as follows: First, choose r nodes uniformly from among all the children of the root. For each successively deeper level, excluding the leaves, choose r nodes uniformly among the nodes at that level that are not descendants of nodes chosen at higher levels. Notice that, since the root of T_i has $f \geq ru$ children, it is always possible to choose enough nodes with this property at each level. Independently, for each of these ru nodes, v , choose a string $w_v \in Z(n_{i+1}, s_{i+1})$ from D_{i+1} and label the leftmost leaf in the h -th subtree of v with the h -th symbol of w_v , for $h = 1, \dots, n_{i+1}$. Label all other leaves of T_i with \perp .

Lemma 5: Suppose that L is a regular language accepted by a deterministic finite automaton M with q states. Suppose (b, k, t, n, s) satisfies the integrality condition, $b \geq 16$, and $2^b \geq 4q$. Let $x \in \Sigma^*$, $A \subseteq [1, n_i]$ with $|A| \geq \alpha n_i$, and $B \subseteq Z(n_i, s_i)$ with $\mu_{D_i}(B) \geq \beta$. Suppose there is a $t-i$ round cell-probe communication protocol, using $m \leq s^k$ memory cells of b bits, that correctly determines whether $x \cdot PRE_j(y) \in L$ for all $j \in A$ and $y \in B$. Then there exist $x' \in \Sigma^*$, $A' \subseteq [1, n_{i+1}]$ with $|A'| \geq \alpha n_{i+1}$, $B' \subseteq Z(n_{i+1}, s_{i+1})$ with $\mu_{D_{i+1}}(B') \geq \beta$ and a $t-i-1$ round cell-probe communication protocol, using m cells of b bits, that correctly determines whether $x' \cdot PRE_{j'}(y') \in L$ for all $j' \in A'$ and $y' \in B'$.

Proof The argument isolates a node v in T_i with the following property: we can fix one round of communication in the original $t - i$ round cell-probe communication protocol to obtain a new $t - i - 1$ round communication protocol that still works well in the subtree rooted at v .

Finding the node v

We examine the behaviour of the Querier during the first round of the original cell-probe communication protocol to find a set of candidates for the node v . For each value of $j \in A$, the Querier sends one of m messages indicating which of the m memory cells it wishes to probe. Colour the j th leaf of T_i with this message.

Since $|A| \geq \alpha n_i$, it follows from Ajtai's Lemma that there exists a level ℓ such that $1 \leq \ell \leq u$ and the fraction of α -dense nodes in level ℓ of T_i is at least $(\alpha - m\alpha^u)/u$. By the integrality condition, $\alpha \leq 1/2$. Furthermore, $u > 6$ and $m \leq s^k = \alpha^{-4kt} = \alpha^{-\frac{u}{2}}$. Therefore

$$(\alpha - m\alpha^u)/u \geq \alpha(1 - \alpha^{\frac{u}{2}-1})/u > 3\alpha/(4u).$$

We now argue that there is a sufficiently large set of candidates for v among the α -dense nodes at level ℓ and a way of labelling all leaves of T_i that are not descendants of these candidates so that the probability of choosing a string in B remains sufficiently large.

Note that in the construction of D_i from D_{i+1} , the r nodes chosen on level ℓ are not uniformly chosen from among all nodes on level ℓ . The constraint that these nodes not be descendants of any of the $r(\ell - 1)$ nodes chosen at higher levels skews this distribution somewhat and necessitates a slightly more complicated argument.

Consider the different possible choices for the $r(\ell - 1)$ nodes at levels $1, \dots, \ell - 1$ of T_i in the construction of D_i from D_{i+1} . By simple averaging, there is some such choice with $\mu_{D'_i}(B) \geq \beta$, where D'_i is the probability distribution obtained from D_i conditioned on the fact that this particular choice occurred. Fix some such choice.

Let S be the random variable denoting the set of r nodes chosen at level ℓ . Since the choice of nodes at higher levels has been fixed, there are certain nodes at level ℓ that are no longer eligible to be in S . Specifically, each of the r nodes chosen at level $h < \ell$ eliminates its $n_{i+1}^{\ell-h}$ descendants at level ℓ from consideration. In total, there are

$$\sum_{h=1}^{\ell-1} r \cdot n_{i+1}^{\ell-h} < 2r \cdot n_{i+1}^{\ell-1}$$

nodes eliminated from consideration at level ℓ . There are $fn_{i+1}^{\ell-1}$ nodes at level ℓ , so the fraction of nodes at level ℓ that are eliminated is less than $2r/f = 2\alpha/(bu^2) \leq \alpha/(4u)$. Thus, of the nodes at level ℓ that have not been eliminated, the subset H of nodes which are α -dense constitutes more than a fraction $3\alpha/(4u) - \alpha/(4u) = \alpha/(2u)$.

We may view the random choice S of the r nodes at level ℓ as being obtained by choosing r nodes randomly, without replacement, from the set of nodes at level ℓ that were not eliminated. Applying Proposition 1 with $\rho = \alpha/(2u)$ and $|S| = r$,

$$Pr[|H \cap S| \leq r\alpha/(8u)] < 2^{-r\alpha/(4u)} = 2^{-b^2/4}.$$

Since $b > 8$, this probability is smaller than $2^{-2b} = \beta/2$. Let E be the event that at least $\lceil r\alpha/(8u) \rceil = \lceil b^2/8 \rceil$ of the elements of S are α -dense. Then $\mu_{D_i''}(B) \geq \beta - \beta/2 = \beta/2$, where D_i'' is the probability distribution obtained from D_i' conditioned on the fact that event E occurred.

Assume that event E has occurred. Then $|H \cap S| \geq \lceil b^2/8 \rceil$. Let V be the random variable denoting the first $\lceil b^2/8 \rceil$ nodes chosen for S that are also in H . By simple averaging, there is some choice for V with $\mu_{D_i'''}(B) \geq \beta/2$, where D_i''' is the probability distribution obtained from D_i'' conditioned on the fact that this particular choice for V occurred. Fix some such choice.

Finally, consider the different possible choices σ for the sequence of labels on those leaves which are not descendants of nodes in V . By simple averaging, there is some choice for σ with $\mu_{D_i^*}(B) \geq \beta/2$, where D_i^* is the probability distribution obtained from D_i''' conditioned on the fact that this particular choice for σ occurred. Fix some such choice.

By construction, the distribution D_i^* is isomorphic to a cross-product of $\lceil b^2/8 \rceil$ independent distributions D_{i+1} , one for each of the nodes in V . Specifically, for each $v \in V$, the string consisting of the concatenation of the labels of the leftmost descendants of v 's children is chosen from D_{i+1} . (All other descendants of v are labelled by \perp .) For $v \in V$ and y chosen from D_i^* , let $\pi_v(y)$ denote the string consisting of the n_{i+1} characters of y labelling the leftmost descendants of the n_{i+1} children of v . Let $B_v = \{\pi_v(y) \mid y \in B \text{ is consistent with } \sigma\}$. Then

$$\beta/2 \leq \mu_{D_i^*}(B) \leq \prod_{v \in V} \mu_{D_{i+1}}(B_v).$$

Hence, there is some $v \in V$ such that

$$\mu_{D_{i+1}}(B_v) \geq (\mu_{D_i^*}(B))^{1/|V|} \geq (\beta/2)^{8/b^2} = 2^{-16/b} \geq 1/2,$$

since $b \geq 16$. Choose that node v .

Fixing a round of communication for each player

Since v is α -dense, there is some message c the Querier may send in the first round such that $|A'|/n_{i+1} \geq \alpha$, where

$$A' = \{j' \in [1, n_{i+1}] \mid \text{the } j'\text{-th child of } v \text{ is coloured } c\}.$$

Recall that the j' -th child of v is coloured c when there is some $j \in A$ on which the Querier sends message c in the first round and the j -th leaf of T_i is a descendant of that child. We fix the message sent by the Querier in the first round to be c .

For each node $v \in V$ and string $y \in B$, let $\lambda_v(y) \in \Sigma^*$ denote the string consisting of the non- \perp characters of y labelling the leaves of T_i that occur to the left of the subtree rooted at v . For each state p of the deterministic finite automaton M , let $B_{v,p}$ denote the set of strings $y' \in B_v$ for which there exists $y \in B$ consistent with σ such that $\pi_v(y) = y'$ and $x \cdot \lambda_v(y)$ takes M from its initial state to state p . Since B_v is the (not necessarily disjoint) union of the q sets $B_{v,p}$, there is a state p' such that

$$\mu_{D_{i+1}}(B_{v,p'}) \geq \mu_{D_{i+1}}(B_v)/q \geq 1/(2q).$$

Fix any function $\gamma : B_{v,p'} \rightarrow B$ so that, for each string $y' \in B_{v,p'}$, $\gamma(y')$ is consistent with σ , $\pi_v(\gamma(y')) = y'$, and $x \cdot \lambda_v(\gamma(y'))$ takes M from its initial state to state p' . In other words, $\gamma(y')$ witnesses the fact that $y' \in B_{v,p'}$.

There are only 2^b different messages the Responder can send. Therefore, there is some fixed message c' for which

$$\mu_{D_{i+1}}(B') \geq 1/(2q2^b) \geq 2^{-2b+1} = \beta,$$

where B' is the set of strings $y' \in B_{v,p'}$ such that, in round one, given the input $\gamma(y')$ and the query c , the Responder sends c' . We fix the message sent by the Responder in the first round to be c' .

Constructing the $t - i - 1$ round protocol

Choose $x' \in \Sigma^*$ to be any fixed word that takes M from its initial state to state p' .

Consider the following new $t - i - 1$ round protocol: Given inputs $j' \in A'$ and $y' \in B'$, the Querier and the Responder simulate the last $t - i - 1$ rounds of the original $t - i$ round protocol, using inputs $j \in A$ and $y = \gamma(y') \in B$, respectively, where j is the index of some leaf in T_i with colour c that is a descendant of the j' -th child of node v . Note that it doesn't matter which colour c leaf in the subtree rooted at the j' -th child of v is chosen. This is because every leaf in this subtree, except the leftmost leaf, is labelled by \perp , so $\text{PRE}_j(y)$ is the same no matter which leaf in the subtree is indexed by j .

It follows from the definitions of A' and B' that, for inputs j and y , the original protocol will send the fixed messages c and c' during round 1. By construction, the new protocol determines whether $x \cdot \text{PRE}_j(y) \in L$.

Since $\pi_v(y) = y'$ and j is the index of a leaf in T_i that is a descendant of the j' -th child of node v , $\text{PRE}_j(y) = \lambda_v(y) \cdot \text{PRE}_{j'}(y')$. Furthermore $x \cdot \lambda_v(y)$ and x' both lead to the same state p' of M , so $x \cdot \text{PRE}_j(y) \in L$ if and only if $x' \cdot \text{PRE}_{j'}(y') \in L$. Thus the new protocol determines whether $x' \cdot \text{PRE}_{j'}(y') \in L$. \square

We now combine Lemma 4 and Lemma 5 in the main technical result.

Theorem 6: Let L be an indecisive regular language accepted by a deterministic finite automaton with q states. Suppose (b, k, t, n, s) satisfies the integrality condition with $b \geq \max(16, 2(|\Sigma| + 1)^q)$ and $s \geq b^{12t}(8kt)^{4t}$. Then there is no t round cell-probe communication protocol for the static (L, n, s) -prefix problem using s^k memory cells of b bits.

Proof Suppose that there is a t round cell-probe communication protocol for the static (L, n, s) -prefix problem using $m \leq s^k$ memory cells of b bits. Since (b, k, t, n, s) satisfies the integrality condition, $b \geq 16$, and $2^b \geq b \geq 2^{q+1} \geq 4q$, we can apply Lemma 5 t times, starting with $A = [1, n]$ and $B = Z(n, s)$, to obtain $x \in \Sigma^*$, $A' \subseteq [1, n_t]$ with $|A'| \geq \alpha n_t$, $B' \subseteq Z(n_t, s_t)$ with $\mu_{D_t}(B') \geq \beta$, and a 0 round cell-probe communication protocol such that the protocol correctly determines whether $x \cdot \text{PRE}_j(y) \in L$ for all $j \in A'$ and $y \in B'$. This implies that $x \cdot \text{PRE}_j(y) \in L$ for all $j \in A'$ and $y \in B'$ or $x \cdot \text{PRE}_j(y) \notin L$ for all $j \in A'$ and $y \in B'$.

Since $\alpha = s^{-1/(4t)}$ and $t \geq 1$, $\alpha^{1+t} \geq s^{-1/2}$. Furthermore, $s \geq b^{12t}(8kt)^{4t}$, $b \geq 16$, and $b \geq 4q$, so

$$\alpha s t = \frac{\alpha s}{(ru)^t} = \frac{s \alpha^{1+t}}{(bu)^{2t}} \geq \frac{s^{1/2}}{b^{2t}(8kt)^{2t}} \geq \frac{b^{6t}(8kt)^{2t}}{b^{2t}(8kt)^{2t}} = b^{4t} \geq b^4 > 12b^3 > 8qb^2.$$

But $n_t \geq s_t > 0$, $b \geq 2(|\Sigma| + 1)^q$, and $\beta = 2^{-2b+1}$. Therefore, by Lemma 4, there exist integers $a, a' \in A'$ and a string $y \in B'$ such that $x \cdot PRE_a(y) \in L$ and $x \cdot PRE_{a'}(y) \notin L$. This is a contradiction. \square

Theorem 7: If L is an indecisive regular language, then for any $n > 0$ there exists $s \in [1, n]$ such that any cell-probe data structure for the static (L, n, s) -prefix problem using $s^{O(1)}$ memory cells of $2^{(\log n)^{1-\Omega(1)}}$ bits requires time $\Omega(\log \log n / \log \log \log n)$ per query.

Proof We show that, for each positive integer k and each positive constant $\epsilon < 1$, there is a constant $\delta > 0$ so that, for each sufficiently large n , there exists $s \in [1, n]$ such that any cell-probe communication protocol using s^k memory cells of $2^{(\log n)^{1-\epsilon}}$ bits requires at least $\delta \log \log n / \log \log \log n$ rounds to solve the static (L, n, s) -prefix problem.

Let k be a positive integer, let $\epsilon < 1$ be a positive constant, and define $\delta = \min(1/8k, \epsilon/3)$. Let n be sufficiently large so that $\delta \log \log n \geq \log \log \log n$, $(\log n)^{\epsilon/3} \geq 16\delta \log \log n \geq 2$, and $2^{(\log n)^{1-\epsilon}} \geq \max(16, 2(|\Sigma| + 1)^q)$, where q is the number of states in the minimal deterministic finite automaton for L . Let $t = \lfloor \delta \log \log n / \log \log \log n \rfloor$, $b = \lfloor 2^{(\log n)^{1-\epsilon}} \rfloor$, and $s = b^{12t} (8kt)^{4t}$. We will argue, using Theorem 6, that no cell-probe communication protocol using s^k memory cells of b bits can solve the static (L, n, s) -prefix problem in t rounds.

Let $u = 8kt$, $r = b^5 u^2$, $f = b^9 u^5$, $n_t = s_t = s/(ru)^t = (b^7 u)^t$, and $n_0 = f^{(u^t-1)/(u-1)} n_t^{u^t}$. Then (b, k, t, n_0, s) satisfies the integrality condition. By Theorem 6, there is no cell-probe communication protocol using s^k memory cells of b bits that solves the static (L, n_0, s) -prefix problem in t rounds.

Observe that the definitions of t and δ imply that $u = 8kt \leq 8k\delta \log \log n / \log \log \log n \leq \log \log n$, so $u^t \leq (\log n)^\delta$. Furthermore, since $2 \leq 16\delta \log \log n \leq (\log n)^{\epsilon/3}$, $b \leq 2^{(\log n)^{1-\epsilon}}$, $\delta \leq \epsilon/3$, and $\epsilon < 1$, it follows that

$$(bu)^{16t} \leq b^{16\delta \log \log n} (\log n)^{16\delta} \leq b^{(\log n)^{\epsilon/3}} 2^{(\log n)^{\epsilon/3}} \leq 2^{(\log n)^{1-2\epsilon/3} + (\log n)^{\epsilon/3}} \leq 2^{(\log n)^{1-\epsilon/3}}.$$

Therefore

$$n_0 < (fn_t)^{u^t} = (b^{7t+9} u^{t+5})^{u^t} < (bu)^{16tu^t} \leq 2^{(\log n)^{1-\epsilon/3} (\log n)^\delta} \leq 2^{(\log n)} = n.$$

Hence, the static (L, n, s) -prefix problem is at least as hard as the static (L, n_0, s) -prefix problem. This implies that there is no cell-probe communication protocol using s^k memory cells of b bits that solves the static (L, n_0, s) -prefix problem in t rounds. \square

Theorem 8: For any indecisive regular language L and any functions $m(s)$ in $s^{O(1)}$ and $m(n)$ in $(\log n)^{O(1)}$, there is a function $N(s)$ such that any cell-probe data structure for the static $(L, N(s), s)$ -prefix problem using $m(s)$ memory cells of $b(N(s))$ bits requires time $\Omega(\sqrt{\log s / \log \log s})$ per query.

Proof Specifically, we will show that for each $k, k' \geq 1$ there is a $\delta > 0$ so that for any sufficiently large s there is an n such that any cell-probe data structure for the static (L, n, s) -prefix problem using $s^{k/2}$ memory cells of $(\log n)^{k'}$ bits requires time at least $\delta \sqrt{\log s / \log \log s}$ per query.

Let $k, k' \geq 1$ be integers and define $\delta = 1/3k'$. For any value of s such that $\delta\sqrt{\log s/\log \log s} \geq 2$, define $t = \lceil \delta\sqrt{\log s/\log \log s} \rceil$, $u = 8kt$, $b = \lceil s^{\frac{1}{12t}}/u^{1/3} \rceil$, $s_0 = b^{12t}u^{4t} \leq s$, $f = b^9u^5$, $n_t = s_t = (b^7u)^t$, and $n = f^{(u^t-1)/(u-1)}n_t^{u^t} < (fn_t)^{u^t} \leq s^{2u^t}$. Then (b, k, t, n, s_0) satisfies the integrality condition.

Now, for s sufficiently large, $u \leq \sqrt{\log s}$, so

$$u^t \leq (\log s)^{t/2} \leq (\log s)^{\frac{1}{6k'}} \sqrt{\log s/\log \log s} = 2^{\frac{1}{6k'}} \sqrt{\log s \log \log s} \text{ and}$$

$$b = \lceil s^{\frac{1}{12t}}/u^{1/3} \rceil \geq \left\lfloor s^{\frac{k'}{4}} \sqrt{\frac{\log \log s}{\log s}} / (\log s)^{1/6} \right\rfloor = \left\lfloor 2^{\frac{k'}{4}} \sqrt{\log s \log \log s} - \frac{1}{6} \log \log s \right\rfloor.$$

Therefore $b/u^{k't} \geq (2 \log s)^{k'}$, for s sufficiently large.

Choose s large enough so that $b \geq \max((2u^t \log s)^{k'}, 16, 2(|\Sigma| + 1)^q)$, where q is the number of states in the minimal deterministic finite automaton for L . Then, by Theorem 6, there is no cell-probe communication protocol using s_0^k memory cells of b bits that solves the static (L, n, s_0) -prefix problem in t rounds. Since $s \geq s_0$, the static (L, n, s) -prefix problem is at least as hard.

Note that $s < (b + 1)^{12t}u^{4t} = (1 + 1/b)^{12t}s_0 < s_0^2$, since $1 + 1/b \leq b$. Hence, $s_0^k \geq s^{k/2}$. Furthermore, $n < s^{2u^t}$, so $(\log n)^{k'} < (2u^t \log s)^{k'} \leq b$. Thus there is no cell-probe communication protocol using $s^{k/2}$ memory cells of $(\log n)^{k'}$ bits that solves the static (L, n, s) -prefix problem in t rounds. \square

Observe that any data structure for the static PREDECESSOR problem can be easily modified to solve the (L, n, s) -prefix problem for the indecisive regular language $L = (0 + 1)^*1$. We thus obtain the following corollaries.

Corollary 9: The static PREDECESSOR problem for a set S from a universe of size n requires $\Omega(\log \log n / \log \log \log n)$ time on any cell-probe data structure using $|S|^{O(1)}$ memory cells of $2^{(\log n)^{1-\Omega(1)}}$ bits each.

A result similar to Corollary 9 was independently shown by Xiao [16].

As noted in the introduction, the static PREDECESSOR problem for a set S from a universe of size $|S|^{O(1)}$ can be solved in constant time in the cell-probe model using only $O(|S|)$ memory cells. On the other hand, the running time of the fusion tree data structure does not depend on the universe size (although the number of bits in each memory cell does depend on the universe size). We obtain the following corollary of Theorem 8.

Corollary 10: The static PREDECESSOR problem for a set S from a universe of size n requires $\Omega(\sqrt{\log |S| / \log \log |S|})$ time on any cell-probe data structure using $|S|^{O(1)}$ memory cells of $(\log n)^{O(1)}$ bits whose time does not depend on n .

4 Lower Bounds for Dynamic Problems

The results of this section are really simply translations of the results of the previous section to the dynamic case, using the translation argument given by Miltersen [13].

The *dynamic* (L, n) -*prefix problem* requires answers to the questions “Is $\text{PRE}_j(x) \in L$?” for $x \in (\Sigma \cup \{\perp\})^n$. For this problem, an additional set of update operations on the input string are permitted. These allow one to replace a particular character in a specified position of that string with a specific element of $\Sigma \cup \{\perp\}$. For example, in the prefix problem corresponding to the PREDECESSOR problem, replacing an occurrence of \perp by an occurrence of 0 or 1 corresponds to an insert and replacing an occurrence of 0 or 1 by an occurrence of \perp corresponds to a delete.

The basic idea of the translation is to observe that dynamic algorithms that have small cost per query and do not run for very long can access only a small number of memory cells from a moderate size set of potential memory cells. Using static dictionary techniques from [8] one can obtain an efficient solution to the static problem by beginning with string in $\{\perp\}^*$, then inserting the *non* – \perp elements needed, one by one, and recording in the dictionary the changes made to the memory. For appropriate choices of parameters one can then use the static lower bounds to derive the following.

Theorem 11: Let L be an indecisive regular language. For any dynamic cell-probe data structure using cells of $2^{(\log n)^{1-\Omega(1)}}$ bits, if updates for the dynamic (L, n) -prefix problem take $2^{(\log n)^{1-\Omega(1)}}$ worst-case time then queries require $\Omega(\log \log n / \log \log \log n)$ worst-case time.

Corollary 12: Any cell-probe data structure for the dynamic PREDECESSOR problem that uses $2^{(\log n)^{1-\Omega(1)}}$ bits per memory cell and $2^{(\log n)^{1-\Omega(1)}}$ worst case time for inserts requires $\Omega(\log \log n / \log \log \log n)$ worst-case time for queries.

The following is a simple observation extending Miltersen’s technique to amortized costs, provided the memory is not too large. The bound $2^{O(b)}$ on the number of memory cells is reasonable; it is the number of different cells that can be accessed when performing indirect addressing.

Theorem 13: Let L be an indecisive regular language. For a dynamic cell-probe data structure using $2^{O(b)}$ memory cells of $b = 2^{(\log n)^{1-\Omega(1)}}$ bits, if updates for the dynamic (L, n) -prefix problem take $2^{(\log n)^{1-\Omega(1)}}$ amortized time, then queries require $\Omega(\log \log n / \log \log \log n)$ worst-case time.

5 Conclusions

We have very nearly resolved the exact complexity of a number of problems related to searching sorted lists. It would be nice to close the remaining gap of $\sqrt{\log \log |S|}$ between the upper and lower bounds. It is difficult to estimate reliably which is closer to the truth, although it seems that our lower bound argument gives away too much, leading us to suspect that the upper bound is the correct answer. One might be inclined to rule out the lower bound based simply on its unusual form. However, as Andersson et al. [4] have recently shown the complexity of the simpler static dictionary problem on the more restricted AC^0 RAM model is the same function as in our lower bound.

Acknowledgements

We are grateful to Peter Bro Miltersen for helpful discussions.

References

- [1] M. Ajtai, M. Fredman, and J. Komlós. Hash functions for priority queues. *Information and Control*, 63:217–225, 1984.
- [2] Miklós Ajtai. A lower bound for finding predecessors in Yao’s cell probe model. *Combinatorica*, 8:235–247, 1988.
- [3] A. Andersson, T. Hagerup, S. Nilsson, and R. Raman. Sorting in linear time? In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 427–436, Las Vegas, NV, May 1995.
- [4] A. Andersson, P. B. Miltersen, S. Riis, and M. Thorup. Static dictionaries on AC^0 RAMs: query time $\Theta(\sqrt{\log n / \log \log n})$ is necessary and sufficient. In *37th Annual Symposium on Foundations of Computer Science*, pages 441–450, Burlington, VT, October 1996. IEEE.
- [5] Arne Andersson. Sublogarithmic searching without multiplications. In *36th Annual Symposium on Foundations of Computer Science*, pages 655–665, Milwaukee, WI, October 1995. IEEE.
- [6] V. Chvátal. Probabilistic methods in graph theory. *Annals of Operations Research*, 1:171–182, 1984.
- [7] Frandsen, P. B. Miltersen, and Skyum. Dynamic word problems. In *Proceedings 34th Annual Symposium on Foundations of Computer Science*, pages 470–479, Palo Alto, CA, November 1993. IEEE.
- [8] M. Fredman and M. Saks. The cell probe complexity of dynamic data structures. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 345–354, Seattle, WA, May 1989.
- [9] M. Fredman and D. Willard. Surpassing the information theoretic bound with fusion trees. *Journal of Computer and System Sciences*, 47(3):424–436, 1993.
- [10] M. Fredman and D. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Journal of Computer and System Sciences*, 48(3):533–551, 1994.
- [11] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 539–550, Chicago, IL, May 1988.
- [12] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 103–111, Las Vegas, NV, May 1995. Journal version.
- [13] Peter B. Miltersen. Lower bounds for Union-Split-Find related problems on random access machines. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 625–634, Montréal, Québec, Canada, May 1994.
- [14] P. Van Emde Boas, R Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Mathematical Systems Theory*, 10:99–127, 1977.

- [15] D. E. Willard. Log-logarithmic worst case range queries are possible in space $\Theta(n)$. *Information Processing Letters*, 17:81–84, 1983.
- [16] Bing Xiao. *New bounds in cell probe model*. PhD thesis, University of California, San Diego, 1992.
- [17] A. C. Yao. Some complexity questions related to distributive computing. In *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing*, pages 209–213, Atlanta, GA, April-May 1979.
- [18] A. C. Yao. Should tables be sorted? *Journal of the ACM*, 28:615–628, July 1981.