6.851: Advanced Data Structures Spring 2007

Lecture 15 — April 9, 2007

Mihai Pătraşcu *Scribe: Ivaylo Riskov*

1 Overview

In the last lecture we considered the problem of finding the predecessor in its static version. We also introduced the idea of communication complexity and message elimination to prove lower bounds.

In this lecture we present some ideas that refine the method of message elimination and make it possible to prove even tighter lower bounds.

2 Problem Description and Lower Bounds

In the static predecessor problem we have n integers that fit in a word size w and the set representation takes $s = n2^a$ space.

Then we have the following lower bound:

$$max \left\{ \begin{array}{l} \log_B n \\ \log_w n \\ \lg \dfrac{w - \lg n}{a} \\ \dfrac{\lg w/a}{\lg \left(\frac{a}{\lg n} \lg w/a \right)} \\ \lg \left(\lg \dfrac{w}{a} / \lg \dfrac{\lg n}{a} \right) \end{array} \right\}$$

Here B is the page size in an external memory model, i.e. when we pay a unit cost to access a page.

The first of these bounds shows that B-trees are optimal. Indeed, this is the only bound where B shows up, so for some regime of the parameters it must be tight.

The second and third bounds are for fusion trees and van Emde Boas queues respectively. There are cases when either of them can be optimal. In particular, the van Emde Boas structure is optimal when space is $n \lg^{O(1)} n$, i.e. $a \approx \lg \lg n$. When space is $n \lg^{O(1)} n$ we get the following lower bound:

$$max \left\{ \begin{array}{l} \log_B n \\ \log_w n \\ \dfrac{\lg w}{\lg (\lg w / \lg \lg n)} \end{array} \right\}$$

Finally, when the space is larger, the fourth and the fifth bounds are better. The fourth bound is good when space is $o(n^2)$ and the last bound is good for smaller space: $a = \Theta(\log n)$.

3 Proof sketch

We are going to do a proof sketch for the case when $w = 3 \lg n$ and $a = O(\lg \lg n)$. We will just outline four key ideas. The details on how they interact can be found in the paper by Pătraşcu and Thorup in [1].

3.1 New model for Communication Complexity

In the last lecture we introduced communication complexity, where the goal was to compute a certain function $f(x_i, y)$ and the two players, Alice and Bob, are allowed to exchange messages of size $h - 1$ bits. In that model, Alice has the queries $x_i, \ldots, x_h$ and Bob has the the data structure and the inputs $x_1, \ldots, x_{i-1}$, $i \in 1, 2, \ldots, h$. We then used round elimination on a protocol with a certain error rate to obtain a lower bound on the number of messages that Alice and Bob need to exchange.

However, this model is not the best one, so we are going to introduce a new model as follows:

1. Alice and Bob reject their inputs with certain probabilities $Pr[\text{Alice accepts}] = \alpha$ and $Pr[\text{Bob accepts}] = \beta$
2. If the input is accepted then Alice and Bob will communicate so that $f(x_i, y)$ is computed correctly.

In effect, we are replacing the round elimination with a round elimination with rejections. Also in the error model we were accumulating error. Here we are successful with a very small probability. However, this will turn out not to be a problem because of the large number of inputs that we have.

Consider Alice's inputs in a trie where the leaves are the inputs and each internal node corresponds to a sequence $x_1, x_2, \ldots, x_i$.

We either accept an input or reject it. Let $\Gamma(v)$ be the set of messages that Alice sends to Bob for input v. $\Gamma(v) = \emptyset$ if the input is rejected and $|\Gamma(v)| = 1$ if v is accepted. We can also define Γ for internal nodes as all the messages sent to Bob for inputs in the subtree of the node.

$$\Gamma(\text{node}) = \bigcup \Gamma(\text{children})$$

.

Bob has all the inputs $x_1, \ldots, x_i$, so he has a path in the trie ending at node a_i. He looks at $\Gamma(a_i)$, uniformly selects a random message, $m(a_i)$, and assumes this is the message that Alice is going to send him. There are two cases:

1. $m(a_i) \in \Gamma(a_{i+1})$. Alice is happy, because she can always come up with a suffix such that the actual message is what Bob guessed. The protocol continues.

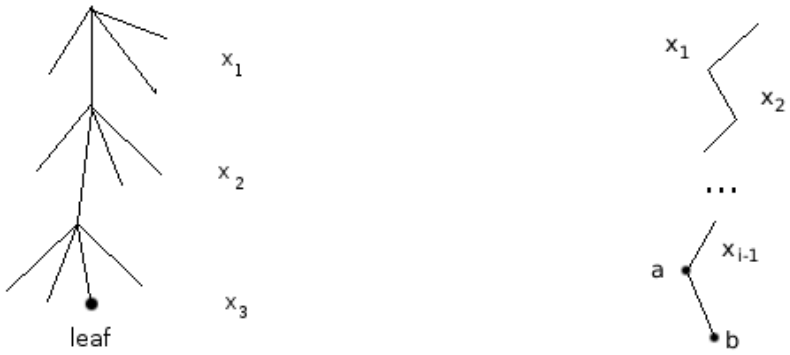

Figure 1: Alice's inputs are in a trie where each leaf is one input. Bob's view of the trie on the right.

2. $m(a_i) \notin \Gamma(a_{i+1})$. Alice rejects.

Lemma 1. *Alice does not reject with probability $\alpha/2h$.*

Let's pick a random leaf v. From the definition of Γ, $Pr[\Gamma(v) = \emptyset] = 1 - \alpha$. Otherwise, $|\Gamma(v)| = 1$.

Then we can bound $|\Gamma(\text{root})| < 2^{h-1}$. Note that there are much more inputs than 2^{h-1}, but once an input is accepted, at most $h-1$ bits can be sent. Since the trie has a depth h (i.e. there are at most h steps), there exists i, such that $|\Gamma(a_{i+1})| \geq \frac{1}{2}|\Gamma(a_i)|$. So for that a_i a random message is good with probability at least $1/2$.

Now we only have to pick the right i, which we can do again by guessing at random. Finally, $Pr[\text{Alice does not reject}] = \alpha \frac{1}{n} \frac{1}{2} = \alpha'$. This suggests that we can eliminate a round at a cost of increasing the rejection probability.

We start with $\alpha = 1$ and at the end we want $\alpha > \left(\frac{1}{2h}\right)^T$, where T is the number of rounds. Similarly for β, $\beta > \left(\frac{1}{2h}\right)^T$. Also $h < w^{O(1)}$ and $T < \lg w$ (because we know that van Emde Boas queues have a lower bound of $\lg w$).

From last lecture we know that eliminating rounds just makes the problem smaller. In the end we need to solve the problem with no communication.

Base Case 2 (Zero communication)**.** *Alice gets random $x \in \{0,1\}^{\lg^3 w}$ and Bob gets colors for the elements $1, 2, \ldots, 2^{lg^3 w}$.*

Alice has $\lg^3 w$ bits of input and has to index into Bob's array. She rejects with probability $2^{-o(\lg^2 w)}$. But even with this tiny probability Alice has to reject lots of inputs. In fact, $2^{\lg^3 w - o(\lg^2 w)}$ possibilities are not rejected, so Bob has to reject the wrong ones otherwise the protocol would not be correct.

The main idea here is that we can tolerate small probabilities. To get a better bound(e,g, $\Omega(\lg w) = \Omega(\lg \lg n)$), however, we need to go beyond communication complexity. The reason is that for $w = 3 \lg n$ and $a = O(\lg \lg n)$ Alice can communicate the bits of the input to Bob in 3 rounds. However, Bob cannot just remember the bits of the first, then of the second and then of the third round. Bob does not have a memory to write down the bits, he has a data structure.

h]

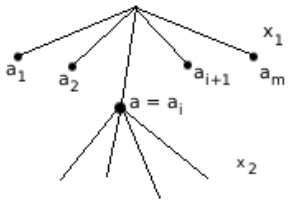

Figure 2: The trie for $f^{(2)}$, there are m possibile inputs for x_1.

3.2 Published Information

Here we assume that Bob is dumb and he cannot mark the cells that Alice probes (we can think of the messages from Alice as cell probes). Instead, we allow Alice to publish some cells that are visible to the world. This published information is accessed for free. In this model, publishing information is equivalent to cell probing, so to eliminate messages from Alice to Bob we just publish the requested cell.

Let's see how to eliminate messages from Bob to Alice. As in last time, we divide the set of integers that Bob has into chunks, each chunk of size w. In the old model we ran the query on all chunks. Now, we change this and instead of having one query, we have w queries that we run on the chunks. This will increase the number of subproblems, k, by w. So to prove the $\Omega(\lg w)$ bound, we just need to examine $f^{(2)}$, i.e. for $h = 2$. At this point, we need to distinguish the h which we use for the bits of information in the protocol and the size of the input: $x_1, \ldots, x_h$.

Claim 3. *If $s = n2^a$ is the size of the space, then we can handle $f^{(2)}$ by publishing $\sqrt{s}$ cells.*

To prove the claim consider again the trie from the previous section and the following two cases:

1. If $\exists i, |\Gamma(a_i)| \leq \sqrt{s}$, then we are happy, because Bob knows x_1 and we just publish $\Gamma(a_i)$.
2. If $\forall i, |\Gamma(a_i)| > \sqrt{s}$, we are still happy. Bob can publish $\sqrt{s} \lg m$ cells. Then with probability $1/m$ these published cells hit $\Gamma(a_i)$ for all i. This implies that with hith probability all of $\Gamma(a_i)$ will be hit.

3.3 Idea 3

The intuition behind this idea is that we can have k queries that query k different data structures and each structure is of size s/k. Then to eliminate a probe from any given data structure we need to publish $\sqrt{s/k}$ cells. This follows from the claim in the previous section. So in total we need to publish $k\sqrt{s/k} = \sqrt{sk}$ to eliminate the probe from all data structures. Here we used that all data structures are independent and we publish $\sqrt{s/k}$ queries from each of them. We can set $k' = \sqrt{sk}$ for the next round. As we progress through the rounds, we can see a pattern for the values of k and s/k. The following table gives the intuition:

Round	k	s/k
0	1	S
1	$\sqrt{s}$	$\sqrt{s}$
2	$s^{3/4}$	$s^{1/4}$

The termination condition is $k < n$, so if s is close to n, we terminate in $\Omega(\lg \lg s)$ rounds. However, $w > 1$ and $\lg w = \lg \lg s$ for $w = \Theta(\lg n)$ (and s close to n). Therefore, we terminate in $\Omega(\lg w)$ rounds.

References

[1] M. Pătraşcu, M. Thorup, *Time-space trade-offs for predecessor search*, Symposium on Theory of Computing, 232-240, 2006

[2] P. Beame, F. Fich, *Optimal bounds for the predecessor problem and related problems*, Journal of Computer and System Sciences, 65(1):38-72, 2002.

[3] A. Andersson, M. Thorup *Tight(er) worst-case bounds on dynamic searching and priority queues*, Symposium on Theory of Computing, 335-342, 2000