6.851: Advanced Data Structures	Spring 2010

Lecture 11 — Mar 11, 2010

Prof. Erik Demaine *Scribe: Jingjing Liu*

1 Overview

In the last lecture we covered static fusion tree. A data structure storering n w-bit integers that supports predecessor and successor queries in $O(\log_w n)$ time with $O(n)$ space.

In this lecture we discuss lower bounds on the cell-probe complexity of the static predecessor problem with constrained space. In particular, we use round elimination technique to prove the preprocessor lower bound in communication model and that the min of van Emde Boas trees and fusion trees is an optimal static predecessor data structure up to loglog factors.

2 Predecessor lower bound results

2.1 The problem

Given a set of n w-bit integers, the goal is to efficiently find predecessor of element x. Observe that having $O(2^w)$ space one can precompute and store all the results to achieve constant query time, we assume $O(n^{O(1)})$ space for our data structures.

The results we are about to discuss are actually for an easier problem: colored predecessor. Each element is colored red or blue. Given query on element x, the goal is to return the color of x's predecessor. Since we can solve colored predecessor problem using predecessor, gives a stronger lower bound for our original problem.

2.2 Results

- *Ajtai-Combinatorica 1988[1]* –Proved the first superconstant bound, $O(w)$; claimed that $\forall w, \exists n$ that gives $\Omega(\sqrt{\lg w})$ query time.
- *Miltersen-STOC 1994[2]* –Rephrased the same proof ideas in terms of communication complexity: $\forall w$, $\exists n$ that gives $\Omega(\sqrt{\lg w})$ query time; $\forall n$, $\exists w$ that gives $\Omega(\sqrt[3]{\lg n})$ query time.
- *Miltersen,Nisan,Safra,Wigderson-STOC 1995[3]*&*JCSS 1998[4]* –Introduced round elimination technique and used it to give a clean proof of the same lower bound.
- *Beame,Fich-STOC 1999[5]*&*JCSS2002[6]*&*manusccript 1994* –Proved two strong bounds: $\forall w, \exists n$ that gives $\Omega(\frac{\lg w}{\lg \lg w})$ query time; $\forall n, \exists w$ that gives $\Omega(\sqrt{\frac{\lg n}{\lg \lg n}})$ query time. Also gave a static data structure achieving $O(\min\{\frac{\lg w}{\lg \lg w}, \sqrt{\frac{\lg n}{\lg \lg n}})\}$, which shows that these bounds are optimal if we insist on pure bound in n&w.

- *Xiao - Ph.D. thesis 1992 at U.C. San Diego[7]* –Independently proved the same lower bound earlier of Beame and Fich.
- *Sen - CCC 2003[8]; Sen, Venkatesh-JCSS2008[9]* –Gave a stronger version of the round elimination lemma that we about to introduce in this lecture, which gives a cleaner proof of the same bounds.
- *Patrascu, Thorup - STOC 2006[10]; SODA2007[11]* –Gave tight bounds for optimal searching predecessors among a static set of integers when $a = \lg \frac{space}{n}$:

$$\Theta(min\{\log_w n, \lg(\frac{w - \lg n}{a}), \frac{\lg \frac{w}{a}}{\lg(\frac{a}{\lg n} \lg \frac{w}{a})}, \frac{\lg \frac{w}{a}}{\lg(\lg \frac{w}{a} / \lg \frac{\lg n}{a})}\}) \quad (1)$$

This trade-off between $n\&w\&space$ shows that given $n \lg^{O(1)} n$ space, the optimal search time is $\Theta(min\{\log_w n, \frac{\lg w}{\lg \frac{\lg w}{\lg \lg n}}\})$. The result also implies that van Emde Boas tree is optimal if $w = O(\lg n)$, and fusion tree is optimal if $\lg w = \Omega(\sqrt{\lg n} \lg \lg n)$

3 Communication Complexity

3.1 Communication complexity view point

We consider the problem in the communication complexity model. Let Alice represent the query algorithm and Bob represent memory. Alice's input is query x; Bob's input is the data structure y. Alice and Bob are only permitted to communicate by sending messages to each other of size at most a and b respectively. Let $a = O(\lg space)$, so using our polynomial space assumption, $a = O(\lg n)$. Let $b = w$, size of a word. The goal is to compute some function $f(x, y)$. In our case, the function is the color of the predecessor. Then #messages exchanged between Alice and Bob is at most twice #probes needed in the cell-probe model. Note, however, that the communication model is much stronger, since it allows both parties to perform arbitrary computation therefore will give a stronger lower bound.

3.2 Predecessor lower bound

Claim: # messages needed in the communication model is $\Omega(\min\{\lg_a w, \lg_b n\})$.

Corollary Beame-Fich-Xiao lower bound:$O(\min\{\frac{\lg w}{\lg \lg w}, \sqrt{\frac{\lg n}{\lg \lg n}})\}$

We have $a = \Theta(\lg n), b = w$, when space is $n^{O(1)}$. The lower bound is largest when $\log_a w = \log_b n$.

$$\log_a w = \log_b n \Rightarrow \frac{\lg w}{\lg \lg n} = \frac{\lg n}{\lg w} \Rightarrow \lg w = \sqrt{\lg n \lg \lg n} \Rightarrow \lg \lg w = \lg \lg n \quad (2)$$

So the bound becomes $\lg_a w = \sqrt{\frac{\lg n}{\lg \lg n}}$. In terms of w, we find $\lg \lg w = \Theta(\lg \lg n)$, so $\lg_b n = \frac{\lg w}{\lg \lg w}$. Therego lower bound is $\sqrt{\frac{\lg n}{\lg \lg n}} = \frac{\lg w}{\lg \lg w}$.

4 Round Elimination

Round elimination can be applied to an abstract communication game (not neccessarily related to the predecessor problem). It gives some conditions under which the first round of communication can be eliminated. To do this, we consider the "k-fold" of an arbitrary function f:

Definition 1. *Let $f^{(k)}$ be a variation on f, in which Alice has the k inputs $x_1, \ldots, x_k$, and Bob has inputs: y, $i \in 1, \ldots k$, and $x_1, \ldots, x_{i-1}$ (note that this overlaps with Alice's inputs). The goal is to compute $f(x_i, y)$.*

Now assume Alice must send the first message. Observe that she must send this message even though she doesn't know i yet. Intuitively, if $a \ll k$, she is unlikely to send anything useful about x_i, which is the only part of her input that matters. Thus, we can treat the communication protocol as starting from the second message.

Lemma 2 (Round Elimination Lemma)**.** *Assume there is a protocol for $f^{(k)}$ where Alice speaks first that uses m messages and has error probability δ. Then there is a protocol for f where Bob speaks first that uses $m-1$ messages and has error probability $\delta + O(\sqrt{a/k})$.*

Intuition If i was chosen uniformly at random (which is the worst case), in Alice's first message the expected number of bits "about x_i" is $\frac{a}{k}$. Bob can guess these bits at random; the probability of guessing all bits correctly is $1/2^{a/k}$, so the probability of failure is $1 - 2^{-a/k}$. Because we are interested in small $\frac{a}{k}$, we have erro increase of $1-2^{-a/k} \approx a/k$. Thus, by eliminating Alice's message, the error probability should increase by about $\frac{a}{k}$. In reality, this intuition is not entirely correct, and we can only bound the increase in the error by $\sqrt{a/k}$, which is often acceptable depending on the application.

5 Proof of Predecessor Bound

Let $t = \#$ cell probes (equivalently, the number of rounds of communication) made by the predecessor algorithm. Our goal is to perform t-round eliminations.

- 1 After t-round eliminations, remaining protocol has no messages, the color of the predecessor must be guessed (assuming $n' \geq 2$), result in $Pr\{success\} \leq \frac{1}{2}$.
- 2 As we perform more eliminations, we are reducing n and w to some n' and w'. We want to increase the probability of error by at most $\frac{1}{3t}$ each time, so that at the end, we still have a nontrivial success probability (at least $\frac{2}{3}$).

So we reach a contradiction.

5.1 Eliminating Alice→Bob

Alice's input has w' bits (initially, $w' = w$). Divide it into $k = \Theta(at^2)$ equal-size chunks $x_1, \ldots, x_k$. Each chunk is of w'/k bits. We want error increase to be $O(1/t)$.

We construct a tree with branching factor $2^{w'/k}$ on the w'-bit strings corresponding to the Alice's possible inputs, which are the elements of the data structure. The tree then has height k. To get lower bound(worst case), constrain n' elements to all differ in ith chunk. Alice and Bob know the structure of the inputs, so Bob knows i, the val the ith chunk, and $x_1, \ldots, x_i$ (because all of Bob's values must start with this common prefix). Thus, when Alice's message is eliminated, the goal changes to query x_i in data structure for ith chunk, w' is reduced to $w'/k = \Theta(w'/at^2)$. An analogy of this data structure is van Emde Boas tree since vEB binary searches on levels to find longest prefix match, reducing w' as it goes. Using the lemma, the error probability increases by $O(\sqrt{a/at^2}) = O(1/t)$, which is exactly what we can afford per elimination.

5.2 Eliminating Bob→Alice

Now that Alice's message is eliminated, Bob is speaking first, so he doesn't know the query's value. Bob's input is n' integers each of size w' bits. Divide the integers into $k = \Theta(bt^2)$ equal chunks of n'/k integers each. Remember that fusion trees could recurse in a set of size $n/w^{1/5}$ after $O(1)$ cell probes. Here, we are proving that after one probe, you can only recurse into a set of size $n/w^{O(1)}$, which gives the same bound for error increase, which is $O(1/t)$.

To get lower bound, constrain input such that ith chunk x_i starts with prefix "i"' in binary. Alice's query starts with some random $\lg k$ bits, which decides which chunk is interesting. If Bob speaks first, he cannot know which chunk is interesting,

So using the lemma, the elimination rises error probability by $O(1/t)$; reduces n' to $n'/k = \Theta(n'/bt^2)$ and w' to $w' - \lg k = w' - \Theta(\lg bt^2)$. As long as w' does not get too small, $w = \Omega(\lg(bt^2))$, this last term is negligible (say, it reduces w' by a factor of at most 2).

5.3 Stopping

Thus, each round elimination reduces n' to $\Theta(n'/bt^2)$ and w' to $\Theta(w'/at^2)$. Further, the probability of error at the end can be made to be at most $\frac{1}{3}$ by choosing appropriate constants.

We stop the elimination when $w' = O(\lg(bt^2))$ or $n' = 2$. If these stop conditions are met, we have proven our lower bound: there were many rounds initially, so we could do enough eliminations to reduce n and w to these small values. Otherwise, we have a protocol which gives an answer with zero messages, and the error probability is at most $\frac{1}{3}$, which is impossible. So we must be in the first case (the stop conditions are met).

Hence, we established a lower bound $t = \Omega(\min\{\lg_{at^2} w, \lg_{bt^2} n\})$. However, because $t = O(\lg n), a \geq \lg n$ and $t = O(\lg w), b = w$, the bases of the logarithms are between a and a^3 and between b and b^3 respectively. Thus, we found $t = \Omega(\min\{\lg_a w, \lg_b n\})$.

References

[1] Miklos Ajtai: *A lower bound for finding predecessors in Yao's cell probe model*, Combinatorica 8(3): 235-247, 1988.

[2] Peter Bro Miltersen: *Lower bounds for union-split-find related problems on random access machines*, Symposium on the Theory of Computing 1994: 625-634.

[3] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, Avi Wigderson: *On data structures and asymmetric communication complexity*, Symposium on the Theory of Computing 1995: 103-111.

[4] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, Avi Wigderson: *On Data Structures and Asymmetric Communication Complexity*, Journal of Computer and System Sciences, 57(1): 37-49 (1998)

[5] Paul Beame, Faith E. Fich: *Optimal Bounds for the Predecessor Problem*, Symposium on the Theory of Computing 1999: 295-304.

[6] Paul Beame, Faith E. Fich: *Optimal Bounds for the predecessor problem and related problems*, Journal of Computer and System Sciences, p.38-72, August 2002.

[7] Bing Xiao: *New bounds in cell probe model*, PhD thesis, University of California, San Diego, 1992.

[8] Pranab Sen: *Lower bounds for predecessor searching in the cell probe model*, IEEE Conference on Computational Complexity 2003, 73-83.

[9] Pranab Sen, Srinivasan Venkatesh:*lower bounds for predecessor searching in the cell probe model*, Journal of Computer and System Sciences 2008, 364-385.

[10] Mihai Patrascu, Mikkel Thorup: *Time-space trade-offs for predecessor research*, Symposium on Theory of computing 2006, 232-240.

[11] Mihai Patrascu, Mikkel Thorup: *Randomization does not help searching predecessors*, Symposium on Discreat Algorithms