6.851: Advanced Data Structures Spring 2012

Lecture 13 — April 5, 2012

Prof. Erik Demaine *Scribes: Karan Sagar (2012), Jacob Hurwitz (2012), Jingjing Liu (2010), Meshkat Farrokhzadi (2007), Yoyo Zhou (2005)*

1 Overview

In the last two lectures, we discussed several data structures for solving predecessor and successor queries in the word RAM model: van Emde Boas trees, y-fast trees, and fusion trees. This establishes an upper bound on the predecessor problem:

$$O\left(\min\left\{\lg w, \log_w n\right\}\right)$$

In this lecture we discuss **lower bounds on the cell-probe complexity of the predecessor problem**. In particular, this bound is for the static problem and if we are constrained to polynomial space. We obtain a lower bound of

$$\Omega\left(\min\left\{\frac{\lg w}{\lg\lg n}, \log_w n\right\}\right)$$

using the round elimination technique in a communication model. This shows that the minimum of van Emde Boas trees and fusion trees solves the static predecessor problem optimally, up to a $\lg\lg$ factor.

The lower bound is particularly elegant because it only relies only on information theory ("Is there any operation at all to do this?"). On the other hand, the upper bounds we've seen have relied on the computer technology available, since they use specific bit manipulations and C-style operations.

2 Historical survey of predecessor lower bound results

2.1 The problem

Given a data structure (a set S of n w-bit integers) and a query (an element x, possibly not in S), the goal is to find the predecessor of x in S as efficiently as possible. Observe that with $O(2^w)$ space we can precompute and store all the results to achieve constant query time; to avoid trivializing the problem, 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. In colored predecessor, each element of S is colored red or blue, and the goal is to return the *color* of the predecessor of x in S. Since we can solve the colored predecessor problem using a solution to the predecessor problem, this gives a stronger lower bound for our original problem.

2.2 Results

Note that earlier bounds do not cover all of the $w - n$ tradeoffs.

- Ajtai [Ajt88] proved the first $\omega(1)$ (that is, superconstant) lower bound, claiming that $\forall w, \exists n$ that gives $\Omega(\sqrt{\lg w})$ query time.
- Miltersen [Mil94] rephrased the same proof ideas more coherently in terms of communication complexity and then showed two results:
  - $\forall w$, $\exists n$ that gives $\Omega(\sqrt{\lg w})$ query time.
  - $\forall n$, $\exists w$ that gives $\Omega(\sqrt[3]{\lg n})$ query time.
- Miltersen, Nisan, Safra, and Widgerson [MNSW95, MNSW98] introduced the round elimination technique and used it to give a clean proof of the same lower bound.
- Beame and Fich [BF99, BF02] proved two strong bounds:
  - $\forall w, \exists n$ that gives $\Omega\left(\frac{\lg w}{\lg \lg w}\right)$ query time
  - $\forall n, \exists w$ that gives $\Omega\left(\sqrt{\frac{\lg n}{\lg \lg n}}\right)$ query time.

  They also gave a static data structure achieving $O\left(\min\left\{\frac{\lg w}{\lg \lg w}, \sqrt{\frac{\lg n}{\lg \lg n}}\right\}\right)$, which shows that the two strong bounds listed above are optimal if we insist on pure bounds (bounds dependent only on n or only on w).
- Xiao [Xia92] independently proved the same two lower bounds as Beame and Fich.
- Sen and Venkatesh [Sen03, SV08] gave a stronger version of the round elimination lemma (covered in this lecture), which gives a cleaner proof of the same bounds from Beame and Fich and from Xiao.
- Patrascu and Thorup [PT06, PT07] gave a tight trade-off between n, w, and space for the static problem. Letting the space be $n \cdot 2^a$, where $a = O(\lg \lg n)$, they found a lower bound of

$$\Theta\left(min\left\{\log_w n, \lg\left(\frac{w - \lg n}{a}\right), \frac{\lg \frac{w}{a}}{\lg\left(\frac{a}{\lg n} \lg \frac{w}{a}\right)}, \frac{\lg \frac{w}{a}}{\lg\left(\lg \frac{w}{a} / \lg \frac{\lg n}{a}\right)}\right\}\right).$$

  For some intuition:
  - The first term looks like fusion trees.
  - The second term is roughly $\lg w$, which is roughly van Emde Boas.
  - The last two terms don't have a good intuition. They look a bit similar to van Emde Boas with the $\lg \frac{w}{a}$, but they improve by some factors when a is large.

  If we have $O(n \operatorname{polylg} n)$ space and $a = O(\lg \lg n)$, then we have these consequences:
  - van Emde Boas trees are optimal for small w: if $w = O(\operatorname{polylg} n)$.
  - Fusion trees are optimal for large w: if $\lg w = \Omega(\sqrt{\lg n} \lg \lg n)$.
  - In between small and large w, we have a bound of $\Theta\left(\min\left\{\log_w n, \frac{\lg w}{\lg \frac{\lg w}{\lg \lg n}}\right\}\right)$.

3 Communication Complexity

3.1 Communication complexity view point

We consider the problem in the communication complexity model. In this generic model, Alice knows a value x and Bob knows a value y. Collectively, they would like to compute some function $f(x, y)$. However, they are limited to a protocol in which they alternate sending messages to each other; furthermore, Alice's messages can only be a bits long, and Bob's messages can only be b bits long. We may expect x and y have many more bits than a and b, so you may have to send multiple messages. We think of this problem in rounds of communication, where one round is Alice talks to Bob and then Bob talks to Alice.

Applying this model to the colored predecessor problem, we can think of Alice as the query algorithm, and she knows the query x. Bob is the memory/RAM, and he knows the data structure y. Together, they want to find $f(x, y)$, which is the answer to the colored predecessor query. Intuitively, a round of a communication is a memory read and Alice wants to "ask" Bob for values from the data structure so she can compute the answer.

Thus, Alice's messages will be address bits, so $a = O(\lg(\text{space}))$. Bob will return the values stored in the data structure, so $b = w$, the word size. Each "round" of communication consists of two messages and corresponds to one probe in the cell probe model. (This is a static problem, so we only have cell probe loads, not writes.) Thus, the number of messages equals twice the number of cell probes. If we can establish a lower bound in the cell probe model, that will imply a lower bound in the word RAM model.

3.2 Predecessor lower bound

Claim: The number of messages (and therefore number of cell probes) needed in the communication model is $\Omega(\min\{\lg_a w, \lg_b n\})$.

Corollary: This implies the Beame-Fich-Xiao lower bound of $\Omega\left(\min\left\{\frac{\lg w}{\lg\lg w}, \sqrt{\frac{\lg n}{\lg\lg n}}\right\}\right)$.

Assuming polynomial space, $a = \Theta(\lg n)$. The lower bound is largest when $\log_a w \stackrel{\Theta}{=} \log_b n$, where $\stackrel{\Theta}{=}$ denotes equality up to a constant factor. Solving, we find:

$$\frac{\lg w}{\lg\lg n} \stackrel{\Theta}{=} \frac{\lg n}{\lg w} \Leftrightarrow \lg w \stackrel{\Theta}{=} \sqrt{\lg n \lg\lg n} \Leftrightarrow \lg\lg w \stackrel{\Theta}{=} \lg\lg n$$

Using this equivalence, we now rewrite $\Omega(\min\{\lg_a w, \lg_b n\})$ as

$$\Omega\left(\min\left\{\frac{\lg w}{\lg a}, \frac{\lg n}{\lg w}\right\}\right) = \Omega\left(\min\left\{\frac{\lg w}{\lg\lg n}, \frac{\lg n}{\sqrt{\lg n \lg\lg n}}\right\}\right) = \Omega\left(\min\left\{\frac{\lg w}{\lg\lg w}, \sqrt{\frac{\lg n}{\lg\lg n}}\right\}\right).$$

This representation makes it easiest to see the Beame-Fich-Xiao lower bound. However, using some of the intermediate steps in the above calculation, we can also rewrite the bound as

$$\Omega\left(\min\left\{\frac{\lg w}{\lg\lg n}, \log_w n\right\}\right),$$

which is the format we presented in the overview at the beginning of lecture. This representation makes it easy to see that the first argument to the min function looks (up to a $\lg \lg$ factor) like van Emde Boas complexity, and the second argument looks like fusion tree complexity.

4 Round Elimination

Let's return to the abstract communication model (not necessarily related to the predecessor problem) to discuss round elimination. Round elimination 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 were chosen uniformly at random (which is the worst case), then in Alice's first message the expected number of bits "about x_i" is 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 a/k, a series expansion shows that $1 - 2^{-a/k} \approx a/k$. Thus, by eliminating Alice's message, the error probability should increase by about 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 still acceptable depending on the application.

5 Proof of Predecessor Bound

Proof sketch: Let t be the number of cell probes, or equivalently, the number of rounds of communication. Our goal is to apply the Round Elimination Lemma $2t$ times to eliminate all the messages. At this point, the color of the predecessor must be guessed (assuming $n' \geq 2$, as defined in 5.1), and so the probability of success is at most $\frac{1}{2}$.

If we can bound the increase in error probability by at most $\frac{1}{6}t$ at each step, then $2t$ applications of the Round Elimination Lemma will increase the error probability from 0 to at most $\frac{1}{3}$. However, this yields a probability of success of at least $\frac{2}{3}$, which is a contradiction.

In other words, *no* algorithm with t cell probes can solve this problem, so t is a lower bound on the expected performance of any static randomized colored predecessor data structure.

5.1 Eliminating Alice→Bob

Alice's input has w' bits (initially, $w' = w$). Divide the input into $k = \Theta(at^2)$ equal-size chunks $x_1, \ldots, x_k$. Each chunk is of $\frac{w'}{k}$ bits. If we can bound the error increase by $O(\sqrt{\frac{a}{at^2}}) = O(\frac{1}{t})$ at each step, then tweaking the constants will give $\frac{1}{6}t$ as desired.

We construct a tree with branching factor $2^{\frac{w'}{k}}$ 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. In the worst case, we can constrain the n' (initially, $n' = n$) elements to all differ in the ith chunk. Alice and Bob know the structure of the inputs, so Bob knows i and the common prefix of all elements $x_1, \ldots, x_{i-1}$. Thus, when Alice's message is eliminated, the goal changes to query x_i in data structure for ith chunk, and w' is reduced to $\frac{w'}{k} = \Theta(\frac{w'}{at^2})$.

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 $O(\sqrt{\frac{a}{at^2}}) = O(\frac{1}{t})$, 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 $\frac{n'}{k}$ integers each. Remember that fusion trees could recurse in a set of size $\frac{n}{w^{1/5}}$ after $O(1)$ cell probes. Here, we are proving that after one probe, you can only recurse into a set of size $\frac{n}{w^{O(1)}}$, which gives the same bound for error increase, which is $O(\frac{1}{t})$.

To get lower bound, constrain the 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,

Using the lemma, elimination raises the error probability by $O(\frac{1}{t})$; reduces n' to $\frac{n'}{k} = \Theta(\frac{n'}{bt^2})$ 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 of eliminations reduces n' to $\Theta(\frac{n'}{bt^2})$ and w' to $\Theta(\frac{w'}{at^2})$. Furthermore, 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 stopping conditions are met, we have proven our lower bound: either there were many rounds initially and we could do enough eliminations to reduce n and w to these small values, or we have a protocol which gives an answer with zero messages. But, this would imply that the error probability is at most $\frac{1}{3}$, which is impossible. Thus, we must be in the first case (the stopping conditions are met).

Hence, we have established a lower bound $t = \Omega(\min\{\lg_{at^2} w, \lg_{bt^2} n\})$. However, because $t = O(\lg n)$ and $a \geq \lg n$, we have $a \leq at^2 \leq a^3$. Likewise, because $t = O(\lg w)$ and $b = w$, we have $b \leq bt^3 \leq b^3$. Thus, we can conclude that $t = \Omega(\min\{\lg_a w, \lg_b n\})$.

6 Sketch of the Proof for the Round Elimination Lemma

We ran out of time to discuss this section in class, so it does not appear in the lecture video.

6.1 Some Information Theory Basics

Definition 3. *$H(x)$, called the entropy of x, is the number of bits needed on average to represent a sample from a distribution of the random variable x. Formally,*

$$H(x) = \sum_{x_0} \Pr[x = x_0] \cdot \lg \frac{1}{\Pr[x = x_0]}$$

Definition 4. *$H(x \mid y)$ is the conditional entropy of x given y: the entropy of x, if y is known:*

$$H(x \mid y) = E_{y_0}[H(x|y = y_0)]$$

Definition 5. *$I(x : y)$ is the mutual or shared information between x and y:*

$$I(x : y) = H(x) + H(y) - H((x, y)) = H(x) - H(x \mid y)$$

$I(x : y \mid z)$ is defined in a manner similar to that of $H(x \mid y)$.

6.2 The Round Elimination Lemma

Call Alice's first message $m = m(x_1, \ldots, x_k)$. Next, we use a neat theorem from information theory to rewrite entropy as a sum which can be thought of as a "chain rule for information":

$$a = |m| \geq H(m) = \sum_{i=1}^{k} I(x_i : m \mid x_1, \ldots, x_{i-1})$$

If i is distributed uniformly in $\{1, \ldots, k\}$, then $E_i[I(x : m \mid x_1, \ldots, x_k)] = \frac{H(m)}{k} \leq \frac{a}{k}$. This is why $\frac{a}{k}$ was an estimate for how many bits of information Bob could learn from the message about Alice's message. Note that we bounded $I(x_i : m \mid x_1, \ldots, x_{i-1})$, so even if Bob already knows $x_1, \ldots, x_{i-1}$ and receives m, he still learns at most $\frac{a}{k}$ bits about x_i.

The prove the lemma, we must build a protocol for f given the assumed protocol for $f^{(k)}$. We can build a protocol $f(x, y)$ as follows:

1. Fix $x_1, \ldots, x_{i-1}$ and i a priori (known to both players) at random.
2. Alice pretends $x_i = x$.
3. Run the $f^{(k)}$ protocol, starting at the second message, by assuming the first message is $m = m(x_1, \ldots, x_{i-1}, \tilde{x}_i, \ldots, \tilde{x}_k)$, where $\tilde{x}_j$ is a random variable drawn from the distribution of x_j. Now the first message does not depend on $x_i = x$ (even x_i is chosen randomly), so Bob can generate it by himself, without any initial message from Alice.

4. Now Alice has some actual x, which she must use as x_i, and almost certainly $\tilde{x}_i \neq x$. But we know that $I(x_i : m)$ is very small, so the message doesn't really depend on x_i in a crucial way. This means that a random message was probably good: Alice can now fix $x_{i+1}, \ldots, x_k$, so that $m(x_1, \ldots, x_{i-1}, \tilde{x}_i, \ldots, \tilde{x}_k) = m(x_1, \ldots, x_{i-1}, x, \ldots, x_k)$, for the desired $x_i = x$.

The last step is the crucial one which also introduces an error probability of $O(\sqrt{\frac{a}{k}})$. This can be proved based on the "Average Encoding Theorem" from information theory. There also exists a more subtle problem that this theorem solves: not only must $x_{i+1}, \ldots, x_k$ exist, so that Bob's random guess for a message is made valid, but their distributions are close to the original distributions, so the error probability δ does not increase too much.

References

[Ajt88] M. Ajtai: *A lower bound for finding predecessors in Yao's cell probe model*, Combinatorica 8(3): 235–247, 1988.

[BF99] P. Beame, F. Fich: *Optimal Bounds for the Predecessor Problem*, Symposium on the Theory of Computing 1999: 295–304.

[BF02] P. Beame, F. Fich: *Optimal Bounds for the predecessor problem and related problems*, Journal of Computer and System Sciences 65(1): 38–72, 2002.

[Mil94] P. Miltersen: *Lower bounds for union-split-find related problems on random access machines*, Symposium on the Theory of Computing 1994: 625–634.

[MNSW95] P. Miltersen, N. Nisan, S. Safra, A. Wigderson: *On data structures and asymmetric communication complexity*, Symposium on the Theory of Computing 1995: 103–111.

[MNSW98] P. Miltersen, N. Nisan, S. Safra, A. Wigderson: *On data structures and asymmetric communication complexity*, Journal of Computer and System Sciences, 57(1): 37–49, 1998.

[PT06] M. Patrascu, M. Thorup: *Time-space trade-offs for predecessor research*, Symposium on the Theory of Computing 2006: 232–240.

[PT07] M. Patrascu, M. Thorup: *Randomization does not help searching predecessors*, Symposium on Discrete Algorithms 2007: 555–564.

[Sen03] P. Sen: *Lower bounds for predecessor searching in the cell probe model*, IEEE Conference on Computational Complexity 2003: 73–83.

[SV08] P. Sen, S. Venkatesh: *Lower bounds for predecessor searching in the cell probe model*, Journal of Computer and System Sciences 74(3): 364–385, 2008.

[Xia92] B. Xiao: *New bounds in cell probe model*, PhD thesis, University of California, San Diego, 1992.