20. Predecessor searching problem. Part II

Lecturer: Meena Mahajan *Scribe: Pranabendu Misra*

20.1 The Predecessor Problem

We are given a universe U of size 2^m and a subset $S \subseteq U$, $|S| = n$. For $x \in U$, define the following functions:

Definition 20.1. $\mathrm{pred}_S(x) = \max\{y \in S | y \leq x\}$.

Definition 20.2. $\mathrm{rank}_S(x) = |\{y \in S | y \leq x\}|$.

Definition 20.3. $\oplus\mathrm{rank}_S(x) = \mathrm{rank}(x) \mod 2$.

Now given S, we wish to answer queries on S. The preprocessing algorithm should store information about S in an appropriate way so that given any $x \in U$, we can find $f_S(x)$ efficiently, where f could be one of the above functions.

Definition 20.4. *A randomized $(s, w, t)_\varepsilon$ storage scheme for $f_S(x)$ consists of*

(1) a deterministic storage algorithm which takes as input $S \subseteq U$ and outputs a data structure T with s cells, each cell w bits long.

(2) a randomized query algorithm which, on input $x \in U$, probes at most t cells in T, and outputs $f_S(x)$ correctly with probability at least $(1 - \varepsilon)$.

All these functions depend on S; henceforth we drop the subscript S in pred_S, rank_S and $\oplus\mathrm{rank}_S$ for convenience. Some departures from last time: we use m for the bit size of an element in the universe. We do not assume that the cell-word-size is m but allow an independent parameter w.

Last time we saw an $(O(n), O(m), O(1))$ deterministic scheme for the dictionary problem using FKS hashing. We also saw an $(O(mn), O(m), O(\log m))$ deterministic scheme for the predecessor problem, using X-tries and the dictionary solution. We also stated without proof that there is an $(O(mn), O(m), \min\left[\frac{\log m}{\log\log m}, \sqrt{\frac{\log n}{\log\log n}}\right])$ deterministic scheme for the predecessor problem.

In this lecture we will show that the upper bound is almost tight:

Theorem 20.5. *For $s \in poly(n)$, $w \in poly(m)$, if there is an $(s, w, t)_\varepsilon$ randomized scheme for the predecessor problem, then $t \in \Omega\left[\frac{\log m}{\log\log m}, \sqrt{\frac{\log n}{\log\log n}}\right]$.*

The references for today's lecture are [Sen03, SV08]. We will actually prove the above theorem for the $\oplus$rank function. Then we will make use of the following observation:

Observation 20.6. *If there is an* $(s,w,t)_\varepsilon$ *scheme for* $\mathrm{pred}(x)$*, then there is an* $(s+O(n), w+O(m), t+O(1))_\varepsilon$ *scheme for* $\mathrm{rank}(x)$*. This is because for each* $y \in S$*, if* y *is the predecessor of* x*, then* $\mathrm{rank}(x) = \mathrm{rank}(y)$*. So given* x *we first find* $\mathrm{pred}(x)$*, and then query a dictionary to find* $\mathrm{rank}(\mathrm{pred}(x))$*. And for each* $y \in S$*, we can use the FKS scheme for the dictionary problem, to store* $\mathrm{rank}(y)$*.*

Similarly, under this hypothesis, $\oplus\,\mathrm{rank}(x)$ *also has an* $(s+O(n), w+O(m), t+O(1))_\varepsilon$ *scheme.*

Let (m,n) denote the size of the universe $|U| = 2^m$ and the size of the subset $|S| = n$. We carry these parameters as subscripts with the function.

To actually prove the theorem for the $\oplus\,\mathrm{rank}_{m.n}$ function, we will consider the communication game associated with $\oplus\,\mathrm{rank}_{m,n}$. Alice has an element $x \in U$, with $x = (x_1, \ldots, x_m)$, each $x_i \in \{0,1\}$. Bob has the subset $S = \{y_1, \ldots, y_n\} \subseteq U$. They wish to determine $\oplus\,\mathrm{rank}_{m,n}(x)$ with respect to S.

Now, given a $(2^a, b, t)_\varepsilon$ scheme for $\oplus\,\mathrm{rank}_{m.n}$, there is a protocol for the communication game which satisfies the following,

(a) Messages from Alice to Bob are a bits long,

(b) Messages from Bob to Alice are b bits long,

(c) Alice begins, and there are $2t$ rounds,

(d) The protocol errs with probability at most ε.

The protocol is simple: Bob runs the preprocessing algorithm and constructs the data-structure T. Alice runs the query algorithm. Whenever she needs to probe a cell, she sends the cell number to Bob, who responds with the contents of that cell in T. The randomness can be private or public; it is required only by Alice, while running the query algorithm.

We call any protocol with these properties a $(2t, a, b)^A_{(\varepsilon,m,n)}$ protocol for $\oplus\,\mathrm{rank}_{m,n}$. A $(2t-1, a, b)^B_{(\varepsilon,m,n)}$ protocol for $\oplus\,\mathrm{rank}_{m,n}$ is a similar $(2t-1)$-round protocol where Bob begins the communication. Note that a protocol for (m,n) is also a protocol for (m', n) for every $m' \le m$.

The lower bound proof proceeds as follows. Suppose we have a $(2t, a, b)^A_\varepsilon$ protocol for $\oplus\,\mathrm{rank}_{m,n}$. Using round elimination we will then show that:

$$\begin{aligned}
& (2t, a, b)^A_\varepsilon \text{ protocol for } \oplus\,\mathrm{rank}_{m.n} \\
\Rightarrow\ & (2t-1, a, b)^B_{\varepsilon + \frac{1}{12t}} \text{ protocol for } \oplus\,\mathrm{rank}_{\frac{m}{k},n} \\
& \text{[eliminate Alice's first message; still OK for slightly smaller universe]} \\
\Rightarrow\ & (2t-2, a, b)^A_{\varepsilon + \frac{1}{6t}} \text{ protocol for } \oplus\,\mathrm{rank}_{\frac{m}{k} - \log l, \frac{n}{l}} \\
& \text{[eliminate Bob's first message; still OK for slightly smaller set]}
\end{aligned}$$

We will show that for $c_1 = 72 \ln 2$, $k = c_1 a t^2$, and $l = c_1 b t^2$, each round elimination adds no more than $1/6t$ to the error.

Consider the following parameters: m is any given value. Choose $n = 2^{\log^2 m / \log\log m}$. Set $c_1 = 72 \ln 2$, and let c_2, c_3 be any constants greater than 1. Choose $a = c_2 \log n$, $b = m^{c_3}$.

Let $t = \frac{\log m}{(c_1+c_2+c_3)\log\log m}$. Choose $k = c_1 a t^2$, $l = c_1 b t^2$. With these parameters, we can verify that:

(1) $\frac{m}{k} - \log l \geq \frac{m}{2k}$.

(2) $m' = \frac{m}{(2k)^t} \in m^{\Omega(1)}$.

(3) $n' = \frac{n}{l^t} \in n^{\Omega(1)}$.

Then, if we repeat round elimination t times, we obtain a $(0, a, b)_{\varepsilon+\frac{1}{6}}$ protocol for $\oplus\,\mathrm{rank}_{m',n'}$ for non-trivial m', n'. For $\varepsilon < \frac{1}{3}$, we get a zero round protocol with error less than $\frac{1}{2}$. But this means that with no information whatsoever about the set S (since there is no communication between Alice and Bob), Alice can guess $\oplus\,\mathrm{rank}(x)$ and be right with probability greater than $1/2$, which is a contradiction.

We now proceed to prove the round elimination theorem. Assume that the constants are chosen as above. Suppose P is a $(2t, a, b)^A_\varepsilon$ protocol for $\oplus\,\mathrm{rank}_{m.n}$. We will convert P into a $(2t-2, a, b)^A_{\varepsilon+\frac{1}{6t}}$ protocol for $\oplus\,\mathrm{rank}_{\frac{m}{k}-\log l, \frac{n}{l}}$.

20.2 Round Elimination: Eliminating Alice's message

We will first convert P into a $(2t-1, a, b)^B_{\varepsilon+\frac{1}{12t}}$ protocol Q for $\oplus\,\mathrm{rank}_{\frac{m}{k},n}$. To do so we will use the randomized version of Yao's lemma which states, $\mathrm{R}_\varepsilon(f) = \max_\mu D^\mu_\varepsilon(f)$ where the protocols D^μ_ε are randomized. We will show that for any distribution μ over (x, S), there is a $(2t-1, a, b)^B_{\varepsilon+\frac{1}{12t}}$ protocol Q that solves $\oplus\,\mathrm{rank}_{\frac{m}{k},n}$ well when the inputs are distributed according to μ. Recall that P works well for all distributions; in particular, it works well for (m, n) distributions that somehow extend μ.

Choose any distribution μ over (x, S) where $|U| = 2^{\frac{m}{k}}$ and $|S| = n$. We first design a protocol $(2t, a, b)^A_\varepsilon$ protocol Q' for $\oplus\,\mathrm{rank}_{\frac{m}{k},n}$ with respect to μ. Then we adapt Q' to obtain Q.

The protocol Q'

Consider a run of the protocol P. Let Alice's input be $x' = x_1, \ldots, x_k$ where x' is broken up into blocks of length m/k, and each block x_i is drawn according to μ. Let M be the first message that Alice sends in the protocol P while using randomness R.

$$\begin{aligned}
I(x' : MR) &= I(x' : R) + I(x' : M|R) \\
&\leq 0 + H(M|R) \qquad \text{(the input } x \text{ and randomness } R \text{ are not correlated)} \\
&\leq H(M) \\
&\leq |M| \quad = \quad a
\end{aligned}$$

Therefore,

$$\begin{aligned}
a &\geq I(x_1, \ldots, x_k : MR) \\
&= I(x_1 : MR) + I(x_2 : MR|x_1) + \ldots + I(x_k : MR|x_1, \ldots, x_{k-1})
\end{aligned}$$

Therefore, there is a block numbered $i \in [k]$ such that

$$I(x_i : MR|x_1, \ldots, x_{i-1}) \leq \frac{a}{k}$$

That is, the first message from Alice and the public randomness together give Bob very little information about the ith block, even if Bob knows the strings in all the preceding blocks. Fix such an i. By definition,

$$E_{x_1=u_1,\ldots,x_{i-1}=u_{i-1}} \left[I(x_i : MR|x_1 = u_1, \ldots, x_{i-1} = u_{i-1}\right] \leq \frac{a}{k}$$

So $\exists u_1, \ldots, u_{i-1}$ such that,

$$I\left(x_i : MR|x_1 = u_1, \ldots, x_{i-1} = u_{i-1}\right) \leq \frac{a}{k}$$

Fix these $u_1, \ldots, u_{i-1}$.

Now we start designing Q'. Alice gets $x \in U = 2^{\frac{m}{k}}$ and Bob gets a set $S \subseteq U$ of size n, where (x, S) are drawn according to μ. To run P, they must *extend* their inputs to look like inputs to P. The idea is to embed x and S into the ith block of suitable chosen longer strings, so as to make the first message almost irrelevant.

Bob extends his set by prefixing each element of S with $u_1 \ldots u_{i-1}$ and suffixing it with zeroes. That is, he constructs the set $S' = \{u_1 \ldots u_{i-1} y 0^{(k-1)\frac{m}{k}} | y \in S\}$.

Alice constructs the element x' by prefixing x with $u_1 \ldots u_{i-1}$ and suffixing it with $k - i$ blocks each chosen according to μ using private randomness. Thus $x' = u_1 \ldots u_{i-1} x x_{i+1} \ldots x_k$, where $x_{i+1}, \ldots, x_k$ are drawn according to μ.

Observe that $\oplus\,\text{rank}_{\frac{m}{k},n}(x, S) = \oplus\,\text{rank}_{m,n}(x', S')$. So Alice and Bob can now run the protocol P to determine $\oplus\,\text{rank}_{\frac{m}{k},n}(x, S)$. This is the $(2t, a, b)^A_\varepsilon$ protocol Q' for $\oplus\,\text{rank}_{\frac{m}{k},n}$.

The protocol Q

Observe that because of the way we constructed the protocol Q', the first message M sent by Alice to Bob contains very little information about x, i.e. $I(x : MR) \leq \frac{a}{k}$. Since M contains so little information about x, Bob might as well replace it with an "average" message. This will introduce some additional error, but we can keep this within bounds using the following:

Theorem 20.7. (Average Encoding Theorem) *Let X, Y be correlated random variables with joint distribution $r_{x,y}$. Let F be the marginal distribution of Y. For any x, let F^x denote the distribution of Y conditioned on the event $X = x$. Then,*

$$\sum_x Pr[X = x] \left\|F^x - F\right\|_1 \leq \sqrt{(2\ln 2) I(X : Y)}$$

Proof. Consider the definitions of these quantities:

$$F(y) = \sum_{x'} r_{x',y}; \qquad F^x(y) = \frac{r_{x,y}}{\sum_{y'} r_{x,y'}}; \qquad Pr(X = x) = \sum_{y'} r_{x,y'}.$$

Define the following distributions on XY:

$$P(x,y) = \Pr[X = x]F^x(y) \qquad\qquad Q(x,y) = \Pr[X = x]F(y)$$

The first distribution P is exactly the joint distribution $r_{x,y}$. The second distribution Q is a product distribution: imagine independent random variables X', Y' distributed according to the marginals, and consider their joint distribution. Therefore,

$$\text{LHS in Theorem } = \|P - Q\|_1 \leq \sqrt{(2\ln 2)D(P||Q)} = \sqrt{(2\ln 2)I(X:Y)}$$

Here, $D(P||Q)$ is the relative entropy or Kullbach-Leibler distance between P and Q. Recall the discussion in Lecture 15, where it was related to the total variation Δ, which is itself half the ℓ_1 distance (Lecture 12). This gives the inequality above. $\square$

Now we define the $(2t-1, a, b)$ protocol Q for $\oplus\text{rank}_{\frac{m}{k},n}$, where (x, S) are drawn according to distribution μ.

Alice gets a string x of $\frac{m}{k}$ bits.

Bob gets a set S of size n.

Bob constructs $S' = \{u_1 \ldots u_{i-1}y0^{(k-1)\frac{m}{k}} | y \in S\}$. Bob then uses public randomness R to construct the "average" message. That is, using public randomness he samples $U_i, \ldots, U_k$ according to μ, and then simulates the protocol P to generate the first message Alice would have sent if her input were $u_1 \ldots\ u_{i-1}U_i \ldots U_k$. We call this the "average" message M'.

Observe that Alice also knows M', because Bob uses public randomness R. Now Alice does a "reverse engineering" of M'. Using private randomness, she samples $V_{i+1}, \ldots, V_k$ according to μ, conditioned on the message being M' and V_i being x. She then constructs $x' = u_1 \ldots u_{i-1}xV_{i+1} \ldots V_k$. This ensures that Alice and Bob now have "consistent" states with input x, S and first message M', and Bob still has very little information about x.

Now Alice and Bob proceed using the protocol Q' (which itself uses P) from the second message onwards.

Calculating the error

Assume Alice's input is x. Consider the following distributions on the set of first messages that can be be sent by Alice. Let F^x be the distribution in protocol Q', and F be the distribution in protocol Q where Bob samples an "average" first message. By the Average Encoding Theorem 20.7, and the choice of i, $u_1, \ldots, u_{i-1}$,

$$\sum_x \Pr[X = x]\,\|F^x - F\|_1 \leq \sqrt{(2\ln 2)I(X:MR)} \leq \sqrt{(2\ln 2)\frac{a}{k}}$$

Hence

$$\begin{aligned}
\Pr[Q \text{ errors }] &= \Pr[Q \text{ errors } | M = M']\Pr[M = M'] + \Pr[Q \text{ errors } | M \neq M']\Pr[M \neq M'] \\
&\leq \Pr[Q \text{ errors } | M = M'] + \Pr[M \neq M'] \\
&= \Pr[Q' \text{ errors }] + \sum_x \Pr[X = x]\Pr[M \neq M' | X = x] \\
&\leq \varepsilon + \sum_x \Pr[X = x]\frac{1}{2}\left\| F^x - F \right\|_1 \\
&\leq \varepsilon + \frac{1}{2}\sqrt{2\ln 2}\sqrt{\frac{a}{k}}
\end{aligned}$$

For a suitable choice of k (at least $72(\ln 2)at^2$), we will get the error to be less than $\varepsilon + \frac{1}{12t}$.

20.3 Round Elimination: Eliminating Bob's message

Now assume we have a $(2t-1, a, b)^B_\delta$ protocol P for $\oplus\text{rank}_{M,N}$, where $M = m/k$ and $N = n$. Following a similar strategy as above, we will convert P into a $(2t-2, a, b)^A_{\delta + \frac{1}{12t}}$ protocol Q for $\oplus\text{rank}_{M - \log l, \frac{N}{l}}$.

Consider any distribution μ on (x, S), where $x \in 2^{M - \log l}$ and $|S| = \frac{N}{l}$.

Now let Bob's input in protocol P be S. Partition S based on the first $\log l$ bits as $S = [1].S_1 \cup \ldots \cup [l].S_l$, where $[i]$ is the representation of i using $\log l$ bits and $[i].S_i = \{[i] \cdot y | y \in S'_i\}$. Assume that the S_i are chosen according to μ. (P works for any distribution of S; in particular, for this distribution.)

Let M be the first message sent by Bob in protocol P while using randomness R. Then,

$$b \geq I(S : MR) = \sum_i I(S_i : MR | S_1, \ldots S_{i-1})$$

So $\exists i$ such that $I(S_i : MR | S_1 \ldots S_{i-1}) \leq \frac{b}{k}$. Fix such an i. By definition,

$$\frac{b}{k} \geq E_{S_1 = s_1 \ldots S_{i-1} = s_{i-1}} I[S_i : MR | S_1 = s_1, \ldots, S_{i-1} = s_{i-1}]$$

So, $\exists s_1, \ldots, s_{i-1}$ such that $I(S_i : MR | S_1 = s_1, \ldots, S_{i-1} = s_{i-1}) \leq \frac{b}{k}$. Fix these sets $s_1, \ldots, s_{i-1}$.

Now the $(2t-1, a, b)^B_\delta$ protocol Q' for $\oplus\text{rank}_{M - \log l, \frac{N}{l}}$ is as follows. Bob and Alice embed their inputs into inputs suitable for protocol P.

Bob gets a set S of size $\frac{N}{l}$. Bob draws sets $S_{i+1} \ldots S_l$ according to μ using public randomness, and constructs $S' = [1].s_1 \cup \ldots \cup [i-1].s_{i-1} \cup [i].S \cup [i+1].S_{i+1} \cup \ldots \cup [l].S_l$.

Alice gets a string x of length $M - \log l$. Alice constructs the string $x' = [i] \cdot x$.

Now observe that $\oplus\text{rank}_{S'}(x') = \oplus\text{rank}_S(x)$. Therefore Alice and Bob run the protocol P on (x', S'). This is the protocol Q' for $\oplus\text{rank}_{M - \log l, \frac{N}{l}}$.

In this protocol, Alice knows the sets $s_1, \ldots, s_{i-1}$ since they are fixed. By choice of the index i and these sets, knowing this and after getting the first message from Bob, she still has very little (at most b/k) information about S. So if the first message is dispensed with and replace with an average message, the error won't increase much. This gives the protocol Q: As before, Alice will sample the average first message M' with public randomness, and Bob will "reverse engineer" the process to sample $S_{i+1} \ldots S_l$ conditioned on M' and S.

To bound the error, as before, use the Average Encoding Theorem. For a suitable choice of l (at least $72(\ln 2)bt^2$), we will get the error to be less than $\delta + \frac{1}{12t}$.

References

[Sen03] Pranab Sen. *Lower bounds for predecessor searching in the cell probe model*. In *Proc. 18th IEEE Conference on Computational Complexity*, pages 73–83. 2003. doi:10.1109/CCC.2003.1214411.

[SV08] Pranab Sen and Srinivasan Venkatesh. *Lower bounds for predecessor searching in the cell probe model*. J. Computer and System Sciences, 74(3):364–385, 2008. (Preliminary version in in *28th ICALP*, 2001 and *18th IEEE Conference on Computational Complexity*, 2003). arXiv:cs/0309033, doi:10.1016/j.jcss.2007.06.016.