Randomized Algorithms assignment 4

Rasmus Stavenuiter, Jacob Harder

Summary

8.2 Random Treaps

Definition 1. *A* treap *is a binary tree with nodes containing pairs of* keys *and* priorities *(distinct numbers) such that it is a*

- full *(all nodes have 0 or 2 children),* endogenous *(values are stored internally)* binary search tree *with respect to the keys (left children has lower keys, right children higher)*
- heap *with respect to the priorities (paths to the root are decreasing in the priorities)*

Part 1

Problem 1

Analogously to the proof of theorem 6, we wish to find a bound on the probability that more than $\frac{3}{4}2^\ell$ keys from $S\setminus\{q\}$ hash to some given ℓ-intervali, I.

As in the proof let X_x indicate whether $h(x) \in I$, where $x \in S \setminus \{q\}$. Then $X = \sum_{x\in S\setminus\{q\}} X_x$ counts the elements different from q that hash to

I. Then

$$\begin{aligned}
\mu = EX = &\sum_{x\in S\setminus\{q\}} EX_x \\
=&\sum_{x\in S\setminus\{q\}} P(h'(x)\in I) \\
=&\sum_{x\in S\setminus\{q\}} \sum_{y\in I} P\left(h'(x)=y\right) \\
<&\sum_{x\in S\setminus\{q\}} 2^{\ell}\left(1/t+1/p\right) \\
<&n2^{\ell}\left(1/t+1/p\right) \\
\leq& n2^{\ell}\left(\frac{1}{t}+\frac{1}{24t}\right) \\
\leq& n2^{\ell}\frac{25}{24t} \\
\leq&\frac{2}{3}2^{\ell}\frac{25}{24}=\frac{50}{72}2^{\ell}
\end{aligned}$$

Hence we get

$$X\geq\frac{3}{4}2^{\ell}\implies X-\mu\geq\frac{3}{4}2^{\ell}-\mu\geq\frac{1}{18}2^{\ell}>\frac{1}{15}\sqrt{2^{\ell}\mu}$$

and using the fourth moment bound we obtain

$$P\left(X\geq\frac{3}{4}2^{\ell}\right)\leq\frac{4}{\left(\frac{1}{15}2^{\ell}\right)^4}=\frac{4\cdot 15^4}{2^{2\ell}}\in\mathcal{O}\left(\frac{1}{2^{2\ell}}\right)$$

Hence the conclusion in the proof of theorem 6 still holds.

Problem 2

By lemma 2 we get that if there is a run of length $r\geq 2^{\ell+2}$ hence for a given run R, the probability that this run is longer than $2^{\ell+2}$ is bounded by $4P_{\ell}$. Throughout the rest of the argument we will take $\ell=\lg(C\lg n)$. Then $2^{\ell+2}=4C\lg n\in\mathcal{O}(n)$.

Now let I be some ℓ-interval. Then as in the proof of theorem 6 let X_x indicate the event $(h(x)\in I)$ for any $x\in[n]$ and $X=\sum_{x\in[n]}X_x$.
Then X counts the number of keys hashing to I and we still have $\mu=EX\leq$

$\frac{2}{3}2^\ell$, and because we are assuming full randomness the X_x are independent poisson trials.
Hence, using Chernoff bounds and the upper bound for EX we get

$$\begin{aligned}
&P(X \geq \frac{3}{4}2^\ell) \\
\leq &P\left(X > (1+\frac{1}{7})\frac{2}{3}2^\ell\right) \\
\leq &\exp\left(\frac{-\frac{2}{3}2^\ell\frac{1}{49}}{4}\right) \\
= &\exp\left(C\lg n\frac{2}{12\cdot 49}\right) \\
= &n^{-C\frac{1}{294}} \leq \frac{1}{4n^{11}}
\end{aligned}$$

for appropriately chosen C. Observe that we can use this version of the Chernoff bound because $\frac{1}{7} \leq 2e-1$.

Now by the observation derived from lemma 2 we have

$$P\left(|R| \geq 2^{\ell+2}\right) \leq 4P_\ell \leq \frac{1}{n^{11}}$$

and hence

$$P\left(\text{some run is longer than } 2^{\ell+2}\right) \leq \sum_R P\left(|R| \geq 2^{\ell+2}\right) \leq n\frac{1}{n^{11}} = \frac{1}{n^{10}}$$

since there can be at most n runs, one for each key.

Hence by taking complements we get

$$P\left(\text{all runs are no longer than } 2^{\ell+2}\right) \geq 1 - \frac{1}{n^{10}}$$

as desired.

Problem 3

Using the same notation as in the proof of theorem 6 we observe that

$$VX_x = \left(1-\frac{2^\ell}{t}\right)\frac{2^\ell}{t} \leq \frac{2^\ell}{t}$$

and since the hash function is 3-independent the X_x's will be pairwise independent so

$$VX = \sum_{x \in S \setminus \{q\}} VX_x \leq n\frac{2^\ell}{t} \leq \frac{2}{3}2^\ell$$

and similar to the proof of theorem 6 we obtain

$$X \geq \frac{3}{4}2^\ell \implies X - \mu \geq \frac{1}{12}2^\ell > \frac{1}{10}\sqrt{2^\ell VX}$$

yielding (by Chebyshev)

$$P\left(X \geq \frac{3}{4}2^\ell\right) \leq P\left(|X-\mu| > \frac{1}{10}2^{\ell/2}\sqrt{VX}\right) \leq \frac{100}{2^\ell} \in \mathcal{O}\left(\frac{1}{2^\ell}\right)$$

So the expected operation time is

$$\mathcal{O}\left(1 + \sum_{\ell=0}^{\lg t} 2^\ell P_\ell\right) = \mathcal{O}\left(1 + \sum_{\ell=0}^{\lg t} 1\right) = \mathcal{O}\left(1 + \lg t\right) = \mathcal{O}\left(\lg n\right)$$

as desired.

Part 2

There is a correspondance between the sorting tree of the *RandQS* algorithm and the random treaps. This correspondance is basically that the choice of pivots in RandQS corresponds to the choice of priorities in a random treap. Both the RandQS tree and the treap sorts the keys from left to right. Which pivot is chosen, then, is simply the one with highest priority, since must have higher priority than all nodes in its subtrees. Similar to the lecture slides let x_i denote the key with the ith lowest priority. Let p_i denotes the priorities. And let $X_{ij} = 1_{\{x_i \text{ is an ancester of } x_i\}}$. Suppose the x_k has priority $-\infty$. Then $X_{ik} = 1 \iff p_i = \max\{p_i, \ldots, p_{k-1}\}$. So $\mathbb{E}(X_{ik}) = P(X_{ik} = 1) = \frac{1}{k-1+1-i} = \frac{1}{k-i}$. So the expected depth of x_k is $\mathbb{E}(\sum_{i<k} X_{ik}) = \sum_{i=1}^{k-1} \frac{1}{d} = H_{k-1}$.

When deleting a key x in a random treap we assign x priority $-\infty$. Then make rotations until x is a leaf. The number of rotations needed is thus exactly the difference in depth between x and a supposed key with priority $-\infty$.

Part 3

We are given a set S of n keys. Similar to the approach in lecture 6, we

Step I Pick a hash function $h : U \to [2n]$, this time by choosing an odd a and having $h(x) = a \cdot x >> (w - b)$. Let $S_i = \{x \in S \mid h(x) = i\}$, $s_i = |S_i|$ and $B = \sum_{i=0}^{n-1} \binom{n}{2}$. If $B \geq n$ we start over, picking another a. Otherwise continue with step II.

Step II For each $i \in [2n]$ pick a hash function $h_i : U \to [2s_i(s_i - 1)]$ until h_i has no collisions on S_i.

Let $b = \lceil \lg(n) \rceil + 1$. Then since $P(h(x) = h(y)) < 2/2^b \leq 1/n$

$$
\begin{aligned}
\mathbb{E}(B) &= \left(\sum_{x,y \in S, x \neq y} P(h(x) = h(y)) \right) \\
&= \frac{n(n-1)}{2} \frac{2}{2^b} \\
&= \frac{n-1}{2} \\
&\leq n/2
\end{aligned}
$$

So $P(B > n) \leq P(B > 2\mathbb{E}(B)) < 1/2$ by Markov. Thus the expected number of trials before finding the right h is 2. And each trial takes linear time. In the second step in we the analogous with $h_i \to [2^{\lg(s_i(s_i-1))+1}]$ obtaining that the expected number of trials before $B_i < 1$ is less than 2. Each trial again taking linear time.

As for the space requirement we have $2n$ for the first step hash function, and $\sum_i 2s_i(s_i - 1) = 2B < 2n$ for the second step set of hash functions. All in all $4n = O(n)$ space.