CSE 6512 - Homework 1 Solutions

Marius Nicolae

November 18, 2011

P1

The probability that an element in T is sampled in S_1 is $p = \frac{|S_1|}{|T|} = \frac{n^2 2^{m/3}}{|T|}$.

Split T into segments of size d. Let $X = B(d, p)$ be a binomial random variable counting how many elements in a region of length d from T are present in S_1. The mean of X is $\mu = dp = \frac{dn^2 2^{m/3}}{|T|}$.

The probability that one of the final partitions is greater than d is the same as the probability that a region of size d contains less than n^2 elements from S_1: $Prob[part > d] = Prob[X < n^2]$. In order to apply the Chernoff bounds we compute the following:

$$(1-\epsilon)\mu = n^2 \Rightarrow 1-\epsilon = \frac{|T|}{d2^{m/3}} \Rightarrow \epsilon = 1 - \frac{|T|}{d2^{m/3}} = \frac{d2^{m/3} - |T|}{d2^{m/3}}$$

We apply the Chernoff bound:

$$\begin{aligned}
Prob[X < n^2] &< \exp\left(-\epsilon^2\mu/2\right) \\
&= \exp\left(-\left(\frac{d2^{m/3} - |T|}{d2^{m/3}}\right)^2 \frac{dn^2 2^{m/3}}{|T|}/2\right) \\
&= \exp\left(-\frac{(nd2^{m/3} - |T|)^2 n^2}{2d|T|2^{m/3}}\right)
\end{aligned}$$

The probability that there exists any part of length d with less than n^2 elements of S_1 is $\leq \frac{|T|}{d} Prob[X < n^2]$. If q is the maximum size of the final parts, then:

$$Prob[q > d] \leq \frac{|T|}{d} \exp\left(-\frac{(nd2^{m/3} - |T|)^2 n^2}{2d|T|2^{m/3}}\right)$$

Now, if we use $d = (1 + n^{-1/3})|T|/2^{m/3}$ we have:

$$
\begin{aligned}
Prob[q > (1+n^{-1/3})|T|/2^{m/3})] &\le \frac{2^{m/3}}{1+n^{-1/3}} \exp\left(-\frac{\left(n(1+n^{-1/3})|T| - |T|\right)^2 n^2}{2(1+n^{-1/3})|T|^2}\right) \\
&< 2^{m/3} \exp\left(-\frac{\left(n(1+n^{-1/3}) - 1\right)^2 n^2}{2(1+n^{-1/3})}\right) \\
&< 2^{m/3} \exp\left(-\frac{n^2(1+n^{-1/3})^2 n^2}{2(1+n^{-1/3})}\right) \\
&= 2^{m/3} \exp\left(-\frac{n^4(1+n^{-1/3})}{2}\right) \\
&< 2^{m/3} \exp\left(-\frac{n^4(1+1)}{2}\right) \\
&= 2^{m/3} \exp\left(-n^4\right) \\
&< 2^{m/3-n^4/2}
\end{aligned}
$$

We have used the fact that $n^{-1/3} < 1, \forall n > 1$ and $e < 2^{.5}$. As long as $m/3 - n^4/2 < -n$ we have the bound required in the problem (*that's the best I could come up with*). A similar result can be obtained for the other inequality, analogously.

P2

We assign a polynomial to each node in each of the two trees, by the following rules:

- Every leaf gets polynomial $P = x_0$
- An internal vertex v at height h having children $v_1, v_2, \ldots, v_k$ gets polynomial $P_v = (x_h - P_{v_1})(x_h - P_{v_2}) \ldots (x - P_{v_k})$

We claim that the two trees are isomorphic if and only if the polynomials at their roots are equal. The left to right implication is immediate: if the trees are isomorphic then the polynomials will be identical by virtue of multiplication being commutative.

If the polynomials are equal, then we can prove by induction that the trees are isomorphic. The base case is trivial: if two trees have polynomial x_0 then they are single node trees and are isomorphic. If the polynomial at the root of both trees is $P_v = (x_h - P_{v_1})(x_h - P_{v_2}) \ldots (x - P_{v_k})$, since x_h does not appear in any of $P_{v_i}, i = 1, k$ it must be the case that both trees have k children which can be paired based on their polynomials. By induction, since the polynomials of the children are equal, the children's subtrees are isomorphic and thus the two trees are isomorphic.

So the problem of checking tree isomorphism has been reduced to checking equality of (at most) degree n multivariate polynomials. This can be done in $\tilde{O}(n)$ as presented in class.

P3

A trivial algorithm is algorithm 1. The worst case runtime is $O(mT_c)$ where T_c is the time needed to check if a graph has a perfect matching,which is the same as the time needed to multiply two matrices.

Algorithm 1 Algorithm P3

```
1. pick edge e = (u, v) ∈ E
remove edge e and nodes u and v from the graph
if there exists a perfect matching for the new graph then
  recursively compute perfect matching M on the remaining graph
  return M ∪ (u, v)
else
  restore graph to initial state
  remove edge (u, v) but not nodes u and v
  go to 1
end if
```

P4

For each submatrix of size m^2, the probability of giving an incorrect answer is $\leq \frac{m^2}{t/\log t}$ where $[1, t]$ is the range out of which we choose prime p. The probability of giving an incorrect answer for any of the $(n-m+1)^2$ submatrices is $\leq \frac{(n-m+1)^2 m^2}{t/\log t}$. The max of $(n-m+1)^2 m^2$ is obtained for $m = n/2$ and is $O(n^4)$. So, if we choose $t = n^{\alpha+4.1}$ then the previous probability is less than $n^{-\alpha}$.

The runtime is $O(n^2)$ using the following observation. Let $B[i, j]$ be the fingerprint of the submatrix having the lower right corner at (i, j) and let $C[i, j]$ be the fingerprint of m contiguous values in row i, ending at position j . Then

$$B[i, j] = (B[i-1, j] - 2^{m^2-m} C[i-m+1, j])2^m + C[i, j](mod\ p)$$

For row i, we only need values of B from row $i-1$, so the extra memory for B is linear. The values of C for row i and $i-m+1$ at column j can be computed on the fly as we scan the current row from left to right, so for C we only add constant memory overhead.

Thus, we can compute the fingerprint of the submatrix ending at (i, j) in constant time from the fingerprint of the submatrix ending at position $(i -$

$1, j)$. Since testing each fingerprint takes $O(1)$ time, the total runtime is $O(n^2)$ (including the cost of fingerprinting the initial submatrices ending at positions in row $m-1$).

P5

a) For every element x in the skip list,

$$Prob\,[level(x) \geq h] = \sum_{i \geq h} p^i \leq \frac{p^h}{1-p}$$
$$\Rightarrow Prob[\exists x \mid level(x) > h] \leq \frac{np^h}{1-p}$$

We want this $\leq n^{-\alpha}$:

$$\begin{aligned}\frac{np^h}{1-p} &= n^{-\alpha} \Rightarrow -(\alpha+1)\log_p n = h + \log_p(1-p) \\ &\Rightarrow h = (\alpha+1)\log_{1/p} n + \log_{1/p}(1-p) \\ &\Rightarrow h = \tilde{O}(log_{1/p} n).\end{aligned}$$

b) The expected number of children for each node is $1/p$ which means the expected runtime of each operation is $(1-n^{-\alpha})O(\frac{1}{p}\log_{1/p} n) + n^{-\alpha}O(n) = O(\frac{1}{p}\log_{1/p} n) = O(\log_{1/p} n)$.

c) In practice, if p is small then the height of the skiplist is small, but the number of children to be scanned at each level increases. Conversely, if p is large, the height increases, but the time at each level is reduced. The minimum value for the function $1/p \log_{1/p} n$ is obtained for $p = 1/2$ which means our initial sampling probability was optimal.

P6

Let H be some random hash family, and let $h \in H$. Let S be a sample of M of size $|S| = s = n$.

$$Prob[h \text{ collides for two values of } S] = \frac{1}{n}$$

$$Prob[h \text{ is perfect for } S] = \left(1 - \frac{1}{n}\right)^{n-1}$$

$$Prob[\forall h \in H, h \text{ is NOT perfect for } S] = \left(1 - \left(1 - \frac{1}{n}\right)^{n-1}\right)^{|H|}$$

$$Prob[\exists S \in M \text{ s.t. there is no perfect } h \in H \text{ for } S] \leq \binom{m}{s} \left(1 - \left(1 - \frac{1}{n}\right)^{n-1}\right)^{|H|}$$

We want to see for what value of $|H|$ this probability is less than 1:

$$\binom{m}{s} \left(1 - \left(1 - \frac{1}{n}\right)^{n-1}\right)^{|H|} < 1$$

$$\Rightarrow \log \binom{m}{s} + |H| \log \left(1 - \left(1 - \frac{1}{n}\right)^{n-1}\right) < 0$$

$$\Rightarrow |H| > \frac{-\log \binom{m}{s}}{\log \left(1 - (1 - \frac{1}{n})^{n-1}\right)}$$

In the last inequality the sign is $>$ because $\log \left(1 - (1 - \frac{1}{n})^{n-1}\right) < 0$.
We know the following facts:

$$\binom{m}{s} < 2^m \Rightarrow \log \binom{m}{s} < m$$

and

$$\left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e}$$

$$\Rightarrow \left(1 - \frac{1}{n}\right)^{n-1} \approx \frac{1}{e(1 - \frac{1}{n})} \approx \frac{1}{e}$$

$$\Rightarrow \log \left(1 - \left(1 - \frac{1}{n}\right)^{n-1}\right) \approx \log \left(1 - \frac{1}{e}\right) \approx -0.199$$

$$\Rightarrow |H| > mc \text{ for some constant } c$$

To sum up, for $|H| = O(m)$, the probability that there is an S for which none of the functions in H is perfect, is < 1. So, the probability there is no such S (meaning $|H|$ is perfect for M) is > 0. Using the probabilistic method, we conclude there exists a perfect hash family, of size polynomial in m.

P7

The size of H is $p-1$. For fixed x and y, $h_a(x) = h_b(y) \Leftrightarrow a(x-y) \equiv in \bmod p$ where $i \in \{1, 2, \ldots, \lfloor \frac{p}{n} \rfloor\}$. So h_a produces collision on x and y only if a is of the form $a = in(x-y)^{-1} \bmod p$. There are $\lfloor \frac{p}{n} \rfloor$ such values, so $\delta(x, y, H) = \lfloor \frac{p}{n} \rfloor \leq \frac{p}{n} = \frac{|H|+1}{n} \leq \frac{2|H|}{n}$. $\square$

P8

We are interested in the probability of getting farther than d positions to the right. Then the probability of being at least d positions away from the origin will be twice that, because the case of going to the left is symmetrical.

Let $X = B(n, 1/2)$ be the event that at any one step we go to the right. If we go a steps towards the right, and $n-a$ towards the left, and at the end we are farther than d away from the origin, towards the right, then $a > (n+d)/2$.

$$\begin{aligned} Prob\,[X > (n+d)/2] &= Prob\left[X > \frac{n}{2}\left(1 + \frac{d}{n}\right)\right] \\ \epsilon &= \frac{d}{n} \Rightarrow^{(Chernoff)} \\ Prob\,[X > (n+d)/2] &< \exp\left(\frac{-d^2}{2n}\right) \end{aligned}$$

We want this $\leq n^{-\alpha}$:

$$\exp\left(\frac{-d^2}{2n}\right) = n^{-\alpha} \Rightarrow \frac{d^2}{2n} = \alpha \ln n \Rightarrow d^2 = 2n\alpha \ln n \Rightarrow d = \sqrt{2n\alpha \ln n}$$

In conclusion $d = \tilde{O}\left(\sqrt{n \log n}\right)$. $\square$

P9

Let $G(V, E)$ be a d-regular graph with m edges and n nodes. Let $R = \max_{i,j} R_{i,j}$ where $R_{i,j}$ is the effective resistance between nodes i and j. We know the following facts:

- The expected cover time $C(G) = O(mR \log n)$
- $\forall i, j \in V, R_{i,j} \leq$ length of the shortest path between i and j
- In a d-regular graph, the diameter is $\leq \frac{n}{d}$
- In a d-regular graph, $m \leq nd$

This means R is no more than $\frac{n}{d}$ and so $C(G) = O(nd\frac{n}{d} \log n) = O(n^2 \log n)$.$\square$