CS6170: Randomized Algorithms
Problem Set #2

Name: Chathurvedhi Talapaneni | Marks: 35
Roll No: CS20B021 | Due: Oct 2, 23:59

Problem 1 — 5 marks

Suppose we are trying to store an m-element set using a Bloom filter. But, unlike what was done in class, we will use functions chosen uniformly at random from a pairwise-independent hash family $\mathcal{H}$. Compute the size of Bloom filter in this case if you want the probability of false positives to be at most δ.

Solution:

Let set be $S \equiv \{x_1, x_2, x_3....., x_m\}$

Let the hash functions be $H_0 = \{h_1, h_2..., h_k\} \in H$

Let the Bloom Filter be an array of size n.

Probability of chosen bit b being zero = $Pr(h_i(x_j) \neq b, for\ all\ i \in [k], j \in [m])$

$\Rightarrow \prod_{h \in H_0}(Pr(h(x_j) \neq b, for\ all\ j \in [m])$

$Pr_m = Pr(h(x_j) \neq b, for\ all\ j \in [m]$ for a construction of n = p, prime number and $h(x_j) = a.x_j + b(modp)$ where a, b chosen at random from $\{0, 1, 2, ...p - 1\}$

Problem 2 **11 marks**

In class, we saw a proof that, w.h.p, the size of any connected component in the Cuckoo graph is $O(\log n)$. In this problem, we will work out an alternate proof of the same using Cayley's formula that is given below.

Theorem 1 (Cayley's formula)**.** *The number of distinct trees on k vertices is k^{k-2}.*

Consider a random graph sampled from $G_{n,p}$ where $p = c/n$ for a constant $c < 1$.

(a) (2 marks) Let X_k be the number of tree components on exactly k vertices for a graph from $G_{n,p}$. A tree component on k vertices will be connected by $k-1$ edges and will be disconnected from the remaining $n-k$ vertices. Show that

$$\mathbb{E}[X_k] = \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{kn - \frac{k(k+3)}{2} + 1}.$$

Solution:

We can find the number of possible tree components with k vertices and the probability of the graph being the actual graph.

The number of such graphs possible:

(Possible choices of k vertices) (Number of distinct trees on k vertices) = $\binom{n}{k} k^{k-2}$

Probability of choosing the given graph:

p^{k-1} for the k chosen vertices

$(1-p)^{(n-k)k + \binom{k}{2} - k}$ for no edge between $n-k$ and k, and no other edge in k vertices

$\Rightarrow p^{k-1}(1-p)^{kn - \frac{k(k+3)}{2} + 1}$

$\therefore \mathbb{E}[X_k] = \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{kn - \frac{k(k+3)}{2} + 1}$.

(b) (2 marks) Show that for $1 \le k \le \sqrt{n}$

$$\mathbb{E}[X_k] \le C \frac{n}{ck^2} e^{(1-c+\ln c)k},$$

for some constant C and large enough n.

Solution: $\mathbb{E}[X_k] = \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{kn - \frac{k(k+3)}{2} + 1}$

$\binom{n}{k} <= \frac{ne}{k}^k$ and $p = c/n$

$\Rightarrow \mathbb{E}[X_k] \le (\frac{ne}{k})^k k^{k-2} (\frac{c}{n})^{k-1} (1 - \frac{c}{n})^{kn - \frac{k(k+3)}{2} + 1}$

$\Rightarrow \mathbb{E}[X_k] \le \frac{n}{k^2} e^k c^{k-1} (1 - \frac{c}{n})^{kn} (1 - \frac{c}{n})^{-\frac{k(k+3)}{2} + 1}$

$1 - x \le e^{-x}$

$\Rightarrow \mathbb{E}[X_k] \le \frac{n}{ck^2} e^{k(1+\ln c)} e^{-kc} e^{\frac{c(k^2+3k-2)}{2n}}$

$e^{\frac{c(k^2+3k-2)}{2n}} \le e^{\frac{c}{2}} = C$ as $1 \le k \le \sqrt{n}$

$\Rightarrow \mathbb{E}[X_k] \le \frac{n}{ck^2} e^{k(1+\ln c)} e^{-kc} C$

$\Rightarrow \mathbb{E}[X_k] \le C \frac{n}{ck^2} e^{(1-c+\ln c)k}$

(c) (2 marks) Using the same expression for $\mathbb{E}[X_k]$, show that

$$\frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} = (n-k)\left(1+\frac{1}{k}\right)^{k-2} p(1-p)^{n-k-2},$$

and in turn that,

$$\frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le \left(1-\frac{k}{n}\right) ce^{1-c(1-k/n)} \left(1-\frac{c}{n}\right)^{-2}.$$

Solution:

Equality:

$\mathbb{E}[X_k] = \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{kn-\frac{k^2}{2}-\frac{3k}{2}+1}$

$\mathbb{E}[X_{k+1}] = \binom{n}{k+1} (k+1)^{k-1} p^k (1-p)^{kn+n-\frac{k^2}{2}-\frac{5k}{2}-1}$

$\frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} = \frac{(n-k)!k!(k+1)^{k-1}p^k(1-p)^{n-\frac{5k}{2}-1}}{(n-k-1)!(k+1)!k^{k-2}p^{k-1}(1-p)^{-\frac{3k}{2}+1}}$

$\therefore \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} = (n-k)\left(1+\frac{1}{k}\right)^{k-2} p(1-p)^{n-k-2}$

Inequality:

$(1+\frac{1}{k})^{k-2} \le (1+\frac{1}{k})^k$, $(1+x) \le e^x$ and $p = c/n$

$\Rightarrow \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le (n-k)\frac{c}{n}(1+\frac{1}{k})^k(1-\frac{c}{n})^{n-k}(1-\frac{c}{n})^{-2}$

$\Rightarrow \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le (1-\frac{k}{n})ce^{\frac{k}{k}}e^{\frac{-(n-k)c}{n}}(1-\frac{c}{n})^{-2}$

$\therefore \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le \left(1-\frac{k}{n}\right) ce^{1-c(1-k/n)} \left(1-\frac{c}{n}\right)^{-2}$

(d) (1 mark) Show that that $xe^{1-x} \le 1$ for $x > 0$, and conclude that

$$\frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le \left(1-\frac{c}{n}\right)^{-2}.$$

Solution:

$\frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le (1-\frac{k}{n})ce^{1-c(1-k/n)}(1-\frac{c}{n})^{-2}$

$\Rightarrow \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le ((1-\frac{k}{n})e^{ck/n})(ce^{1-c})(1-\frac{c}{n})^{-2}$

$e^{ck/n} \le e^{k/n}$

$\Rightarrow \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le ((1-\frac{k}{n})e^{k/n})(ce^{1-c})(1-\frac{c}{n})^{-2}$

$xe^{1-x} \le 1$

$\Rightarrow \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le (1)(1)(1-\frac{c}{n})^{-2}$

$\therefore \frac{\mathbb{E}[X_{k+1}]}{\mathbb{E}[X_k]} \le \left(1-\frac{c}{n}\right)^{-2}$

(e) (4 marks) Using the above, argue that the maximum size of a tree component in G is $O(\log n)$ with probability $1 - o(1)$.

Solution:
Using the result from part (b):
$\mathbb{E}[X_k] \leq C\frac{n}{ck^2}e^{(1-c+\ln c)k}$
We take $k = t\ln n$
$\Rightarrow \mathbb{E}[X_k] \leq C\frac{n}{ct^2(\ln n)^2}e^{(1-c+\ln c)t\ln n}$
If $t = 1/(c - \ln c - 1)$ and $C_0 = C\ (ct^2)$, We obtain:
$\Rightarrow \mathbb{E}[X_k] \leq C_0\frac{n}{(\ln n)^2}e^{-\ln n}$
$\Rightarrow \mathbb{E}[X_k] \leq C_0/(\ln n)^2$
Now we use Markov's Inequality:
$Pr(X_k \geq 1) \leq \mathbb{E}[X_k]/1$
$\Rightarrow Pr(X_k \geq 1) \leq C_0/(\ln n)^2$
$\therefore$ Probability of Max Component Size is $t\ln n \equiv O(\log n)$:
$Pr(X_k \leq 1) \geq 1 - C_0/(\ln n)^2 \equiv 1 - o(1)$

Problem 3 — 11 marks

Collaborator : Sooraj Srinivasan

Consider the scenario of n autonomous agents in a distributed setting vying for a resourcei, say a printer on a network. Assume that there are n copies of the resource available, but an agent will be served by a copy of the resource if it is the only agent that has chosen that instance of the resource. If there are multiple agents that choose the same copy, then that copy gets blocked and the agents will have to wait for the next round. Our goal is to understand the number of rounds before all n agents get served.

Let us model this as a balls into bins process. Here the agents are the balls and the copies of the resource are the bins. In the first round, n balls are thrown independently and uniformly at random into n bins. After round i, we discard all balls that fell into a bin by itself in round i. We continue with the remaining balls in a similar fashion for round $i+1$, where they are thrown independently and uniformly at random into n bins.

(a) (2 marks) If there are b agents waiting to be served at the start of a round, what is the expected number of agents remaining at the start of the next round?

Solution: The probability of an agent successfully accessing the resource is the probability that he falls in the bin alone.
Thus with b balls and n bins, the probability of a given ball being alone is if every other ball falls into another bin
$Pr(Chosen\ ball\ is\ discarded) = (1 - 1/n)^{b-1}$
$Pr(Chosen\ ball\ remains) = 1 - (1 - 1/n)^{b-1}$
$\mathbb{E}[Number\ of\ balls] = b(1 - (1 - 1/n)^{b-1})$

(b) (4 marks) Suppose that in every round the number of agents that are served is exactly the expected number. Show that all the balls would be served in $O(\log \log n)$ rounds.

Solution:

Let X_i be the number of balls after i iterations. $X_0 = n$

From the above we can derive that:

$\frac{X_{i+1}}{X_i} = (1 - (1 - 1/n)^{X_i - 1})$

Let us take $(1 - 1/n)^{X_i - 1} \geq (1 - (1/n)(X_i - 1))$

$\frac{X_{i+1}}{X_i} \leq (1 - (1 - (X_i - 1)/(n))$

$X_{i+1} \leq {X_i}^2/n$

Repeat from $i = k \ to \ t + k$

$X_{t+k} \leq {X_k}^{2^t}/n^{2^t - 1}$

We can use k = 1, $X_1 = n(1 - (1 - 1/n)^{n-1})$

$(1 - (1 - 1/n)^{n-1}) = (1 - (1 + 1/(n - 1))^{1-n}) \leq (1 - 1/e) = 1/c_0$

$X_1 \leq n/c_0$

$\Rightarrow X_{t+1} \leq n/{c_0}^{2^t}$

$n/{c_0}^{2^t} = 1$ when $t = O(\log \log n)$ and k = 1

$\therefore t = O(\log \log n)$

(c) (5 marks) Use the Poisson approximation to show that there is a constant c such that all the agents will be served within $c \log \log n$ rounds with probability at least $1 - o(1)$.

Solution:

Let us use the result X_{t+k} from 4(b)

We know X_i is a binomial r.v $\equiv Bin(X_{i-1}, p)$, where $p = (1 - (1 - 1/n)^{X_{i-1} - 1})$

We can convert this using Poisson Approximation as $X_i \equiv Poi(X_{i-1}p)$ or $Poi(\mu)$

Also for $\mathbb{E}[X_{t+1}] \leq n/c_0^{2^t}$, $c_0 = e/(e-1)$, $\mu = \lambda \leq n/c_0^{2^t}$

Let us take the case where $X_{t+k} \geq 1$

$Pr(X_{t+1} \geq 1) \leq e^{1-\lambda}\lambda \leq e\lambda$

Using $\lambda \leq n/{c_0}^{2^t}$

$Pr(X_{t+1} \geq 1) \leq en/c_0^{2^t}$

If we take $t = \log_2 \log_{c_0} n^2$

$Pr(X_{t+1} \geq 1) \leq k/n$ where k is some constant

$Pr(X_{t+1} < 1) \geq 1 - k/n \equiv 1 - o(1)$

After $\log_2 \log_{c_0} n^2 = (\log_2 e) \log \log n + c$ steps

We can get the probability that all agents will be served in $c \log \log n$ w.p $\geq 1 - k/n \equiv 1 - o(1)$ where k is a constant

Problem 4 **8 marks**

In this problem, our goal is to devise an algorithm for a packet routing problem on a connected undirected graph G. We want to route N packets whose source, destination, and the exact route through the graph G is given. In each time-step, the packet can either traverse an edge or wait at a node. Furthermore, at most one packet can traverse an edge at a given time-step.

A *schedule* for a set of packets specifies the timings for those packets, i.e. it specifies which packet should stay at a node and which should move for every time step. Our goal is to design a schedule that minimizes the total time and the maximum queue size to route all packets to their destinations. We will denote by c, the congestion in the network, which is the maximum number of packets that must traverse a single edge in the network throughout the entire course of routing. By d, we denote the maximum distance travelled by any packet.

(a) (4 marks) First consider the following *unconstrained schedule* where multiple packets are allowed to pass through an edge during one time-step: For a constant α, choose an integral delay independently and uniformly at random from the interval $[1, \lceil \alpha c/\log(Nd) \rceil]$ for each packet. If the delay is x, then the packet stays at the source for x time steps, and then gets routed on its path without any delay at any of the intermediate nodes.

Show that in this unconstrained schedule, the probability that more than $O(\log Nd)$ packets pass through any edge at any given time-step is at most $1/(Nd)$ for a sufficiently large α.

Solution:

Let us take an edge and consider the extreme case where the number of packets passing through the network is c.

Let $y_1, y_2, \ldots.., y_c$ be the time instants from the start of the packet at its source, it goes over the given edge.

And let the delay be X_i for the i^{th} packet and the X_i is randomly chosen from $[1, p]$, $p = \lceil \alpha c/\log(Nd) \rceil$

Let Z_i be the R.V such that $Z_i = 1$ if the packet i crosses the edge at a given time instant, else o

$\mathbb{E}[Z_i] = Pr(y_i + X_i = t) = 1/p$

All Z_i are independent $Ber(1/p)$, $Z = \sum Z_i$

$E[Z] = c/p = \log Nd/\alpha$

Using Chernoff-Hoeffding Bounds and bound with $k \log Nd$:

$Pr(Z \geq (1+\delta)(\mathbb{E}[Z])) \leq exp(-\delta^2(\mathbb{E}[Z])/3)$

$Pr(Z \geq \frac{1+\delta}{\alpha}(\log Nd)) \leq exp(-\delta^2(\log Nd)/(3\alpha)) = 1/(Nd)^{\delta^2/(3\alpha)} \leq 1/(Nd)$

(b) (4 marks) Use the unconstrained schedule from Part (a) to devise a randomized algorithm that, with high probability, produces a schedule of length $O(c + d \log(Nd))$ using queues of size at most $O(\log(Nd))$ such that at most one packet crosses an edge at every time-step.

Solution:

We can use the idea in part (a) where the delay(X_i) can be taken randomly

$X_i \in [1, \lceil \alpha c/\log(Nd) \rceil]$.

From 4(a) we can say that the queue sizes simply represent the number of packets willing to process along an edge at any given time instant is greater than $O(\log Nd)$ with probability at most $1/(Nd)$

From the original case with no restraints we have the entire time of a given packet i $= X_i + k_i$, where k_i is length of path and X_i is delay

$\leq \alpha c/\log(Nd) + d$

With the constraint of one packet moving at a time, each time step in original case can be as long as the queue at the edge $\leq O(\log(Nd))$

$therefore$ We take the worst case $\Rightarrow$ Schedule $= (d + \alpha c/\log(Nd))(O(\log(Nd))) = O(c + d \log(Nd))$

And the probability of this event is every edge $\geq O(\log(Nd))$ packet going over the edge w.p at most $1/(Nd))$

Thus probability of schedule being $O(c + d \log(Nd))$ is $\leq d(1/(Nd)) = 1/N$