CS6170: Randomized Algorithms
Problem Set #2

Name: Your name | Marks: 35
Roll No: Your roll number | 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: Type your solution here.

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: Type your solution here.

(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: Type your solution here.

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

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

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: Type your solution here.

(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: Type your solution here.

(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: Type your solution here.

Problem 3 — 11 marks

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: Type your solution here.

(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: Type your solution here.

(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: Type your solution here.

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: Type your solution here.

(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: Type your solution here.