CS 577: Introduction to Algorithms Homework 7 Solution

Problems

1. **(Worth: 2 points. Page limit: 1 sheet; 2 sides)**

   (a) We have n users on a P2P network who want to pick *distinct* IDs in a distributed fashion. Each user independently picks one uniformly random b bit number. Show that when $b \geq 6 + 2\log n$, the probability of the users picking distinct IDs is at least 0.99.

   SOLUTION: Let A be the event that some users pick duplicated number. Let A_{ij} be the event that user i and user j pick the same number. Then we can upper bound $P(A)$ by:

$$\begin{aligned} P(A) &= P(\cup_{i \neq j} A_{ij}) \\ &\leq \sum_{i \neq j} P(A_{ij}) \\ &= \binom{n}{2} \frac{1}{2^b} \\ &= \frac{n(n-1)}{2^b \cdot 2} \end{aligned}$$

   where the second line is the union bound, the third line holds since for all $i \neq j$, $P(A_{ij})$ should be the same: user i can pick anything, but j has to choose whatever i has chosen, which gives $\frac{1}{2^b}$. The problem boils down to counting how many A_{ij} exist: exactly $\binom{n}{2}$.
   Since $b \geq 6 + 2\log n$, we have:

$$P(A) \leq \frac{n(n-1)}{2^b \cdot 2} \leq \frac{n(n-1)}{2^6 \cdot 2^{2\log n} \cdot 2} \leq \frac{n^2}{2^6 \cdot n^2 \cdot 2} \leq 0.008$$

   Finally this gives:

$$P(\text{No collision}) = 1 - P(A) \geq 1 - 0.008 \geq 0.992$$

   (b) We have n users with distinct IDs on a P2P network, who want to elect a leader in a distributed fashion. Each user independently picks a uniformly random b-bit number, and the leader is determined to be the user with the smallest number. Show that when $b \geq 8 + \log n$, the probability that a unique leader is chosen is at least 0.99.

   SOLUTION: Let A be the event in which a unique leader is elected. Let A_i denote the event that the unique leader picks the number i, where $i \in \{1, 2, \ldots, 2^b - 1\}$. Then we get the following equations, these are explained below:

$$\begin{aligned} P(A) &= P(\cup_{i=1}^{2^b-1} A_i) \\ &= \sum_{i=1}^{2^b-1} P(A_i) \\ &= \sum_{i=1}^{2^b-1} n \cdot \frac{1}{2^b} \cdot \left(1 - \frac{i}{2^b}\right)^{n-1} \\ &= \frac{n}{2^b} \sum_{i=1}^{2^b-1} \left(\frac{i}{2^b}\right)^{n-1} \\ &= \frac{n}{2^{bn}} \sum_{i=1}^{2^b-1} i^{n-1} \end{aligned} \tag{1}$$

The first equality is just by the definition of A and the A_i's. The second equality follows by noting that the A_i's are disjoint events. Let us look at the third equality: since the leader uniquely picks number i, he or she only has one choice out of the 2^b candidates, that is how we get the term $\frac{1}{2^b}$. The rest of the users have to pick a number bigger than the one that the leader selected. Since the leader selected number i, the others have pick one of $(2^b - i)$ numbers. So the probability that the rest of the users (there are $n-1$ of them) do not pick the same number as the leader does is $\left(1 - \frac{i}{2^b}\right)^{n-1}$. Since each of the n users could be the leader, that is why we multiply n in the beginning.
Recall that $\sum_{x=m}^{n} p(x) \geq \int_{m-1}^{n} p(x)dx$. In particular, let $m = 1$, $n = 2^b - 1$, and $p(x) = x^{n-1}$, we have:

$$\sum_{i=1}^{2^b-1} i^{n-1} \geq \int_0^{2^b-1} i^{n-1} di = \frac{i^n}{n}\bigg|_{i=0}^{2^b-1} = \frac{(2^b-1)^n}{n}$$

substitute this result back to Eq. 1, we get:

$$P(A) \geq \frac{n}{2^{bn}} \frac{(2^b-1)^n}{n} = \left(1 - \frac{1}{2^b}\right)^n$$

Substitute $b \geq 8 + \log n$, we have:

$$\left(1 - \frac{1}{2^b}\right)^n \geq \left(1 - \frac{1}{2^8 n}\right)^n$$

From here on we can proceed in two ways:
(i) Use $(1 - \frac{1}{x})^x \geq 0.25$ for $x \geq 2$ (as seen in Section 1), in particular, set $x = 2^8 n = 256n \geq 2$, we get:

$$\left(1 - \frac{1}{2^8 n}\right)^n = \left(\left(1 - \frac{1}{2^8 n}\right)^{256n}\right)^{\frac{1}{256}} \geq 0.25^{1/256} > 0.994$$

(ii) Use $(1 + x) \geq e^{\frac{x}{1+x}}$ (as seen in Section 2), to get:

$$\left(1 - \frac{1}{256n}\right)^n \geq e^{(\frac{-\frac{1}{256n}}{1-\frac{1}{256n}})\cdot n} = (\frac{1}{e})^{\frac{n}{256n-1}} = (\frac{1}{e})^{\frac{1}{256-\frac{1}{n}}} \geq (\frac{1}{e})^{\frac{1}{255}} > 0.996$$

2. **(Worth: 3 points. Page limit: 1 sheet; 2 sides)**

(a) You are given a circle of unit circumference. You pick k points on the circle independently and uniformly at random and snip the circle at those points, obtaining k different arcs. Determine the expected length of any single arc.

(Hint: Note that the length of each arc is identically distributed, so each has the same expectation. What is the sum of their expectations?)

SOLUTION: Let X_i be the random variable that represents the length of the i^{th} arc. Use the hint, we know $X_1, \dots, X_k$ are identically distributed, and they have the same expectation:

$$\mathbb{E}[X_1] = \mathbb{E}[X_2] = \dots = \mathbb{E}[X_k]$$

Also by the linearity of expectation, we have:

$$\sum_{i=1}^{k} \mathbb{E}[X_i] = \mathbb{E}[\sum_{i=1}^{k} X_i] = 1$$

where the second equality is due to the fact that the lengths of k arcs would sum up to the circumference of the circle. This tells that $\mathbb{E}[X_i] = \frac{1}{k}$, for all $i \in \{1, \dots, k\}$.

We can also calculate this value explicitly. This approach is attached in the appendix, you're not required to understand this.

(b) You are given a sorted circular linked list containing n integers, where every element has a "next" pointer to the next larger element. (The largest element's "next" pointer points to the smallest element.) You are asked to determine whether a given target element belongs to the list. There are only two ways you can access an element of the list: (1) to follow the next pointer from a previously accessed element, or (2) via a given function RAND that returns a pointer to a uniformly random element of the list.

Develop a randomized algorithm for finding the target that makes at most $O(\sqrt{n})$ comparisons in expectation and always returns the correct answer.

(Hint: Your algorithm will perform some random accesses and some amount of linear search. Use part (a) to analyze the number of steps in the linear search.)

SOLUTION: There are two ways to access an element: with RAND, and by linear search from some element that has already been accessed. Suppose we access k elements using RAND. Once this is done the only option left is to choose one of these elements and begin a linear search until we reach the next element returned by RAND. Intuitively, there are n/k elements on average between two elements accessed by RAND. So we get that the total number of accesses is $k + n/k$ in expectation. We will show this formally later. We can balance this out by having $k = \sqrt{n}$.

Algorithm:

Let t be the element we look for. Choose $\sqrt{n}$ elements of the circularly linked list by making calls to RAND.
Order them in increasing order to form a list of elements $e_1, e_2, \dots e_k$.
Find e_i such that $e_i \leq t < e_{i+1}$.
If this is not possible, $t < e_1$ or $t \geq e_n$, so t lies between e_n and e_1 if it exists. Thus use $e_i = e_n$.
Starting at e_i follow the circularly linked list until e_{i+1}, returning true if t is found.

Analysis:

We access $\sqrt{n}$ elements in the first step by making calls to RAND. Then we check potentially all elements from e_i to e_{i+1}. So we would like to determine the expected number of elements between e_i and e_{i+1}.

From (a), we know that if we choose k point on a circle with unit circumference uniformly at random, then the expected length of any arc is $\frac{1}{k}$. In this problem, we have a circle with circumference of n, by choosing k points, we would have expected length of any single arc $\frac{n}{k}$. Setting $k = \sqrt{n}$, we have the expected length of any arc is $\frac{n}{k} = \frac{n}{\sqrt{n}} = \sqrt{n}$. The expected length of any arc represents the expected number of elements between e_i and e_{i+1}. Hence, in the worst case, we compare t to every element between e_i and e_{i+1}, which is $\sqrt{n}$ comparison in expectation.

1 Appendix

Here we give an explicit calculation for problem 2a.

Remark 1 *You are not required to understand this approach.*

First notice that if we snip the circle in the first picked point, we can "unroll" the circle into an unit length interval, and the problem becomes "calculate the expected length of intervals created by $k-1$ points on $[0,1]$".

Since each X_i's are identical, we just use X to represent the random variable. Recall $\mathbb{E}[X] = \int xp(x)dx$, where $p(x) = P(X = x)$ is the PDF (probability density function). To get PDF, we need CDF(cumulative distribution function). Let $F(x) = P(X \leq x)$ be the CDF. This is a rather hard term to calculate, as there are many possibilities that the length of certain interval is less than x: if any one of the $k-1$ points fall within a $x-$neighborhood, this event becomes true. So instead we calculate:

$$P(X \geq x) = (1-x)^{k-1} \Longrightarrow P(X \leq x) = 1 - (1-x)^{k-1}$$

Then we have:

$$p(x) = P(X = x) = \frac{dF(x)}{dx} = (k-1)(1-x)^{k-2} \tag{2}$$

Then we finally have:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)dx = \int_0^1 xp(x)dx = \int_0^1 x(k-1)(1-x)^{k-2}dx = \frac{1}{k}$$

The last equality can be calculated using integration by parts.

Remark 2 *Notice Eq 2 isn't always true. This derivation only works if CDF $F(x)$ is absolutely continuous (which I guess is true in most of the cases we would see in this class). The more general PDF is defined via Radon-Nikodym derivative, which you don't need to worry about for this class.*