CS5330: Randomized Algorithms

Problem Set 2

Due: January 29, 6:30pm

Instructions. The *exercises* at the beginning of the problem set do not have to be submitted—though you may. The first three exercises should be easy review. The remaining two exercises are more interesting and it is recommended that you do them. There are three problems to submit.

- Please submit the problem set on IVLE in the appropriate folder. (Typing the solution using latex is recommended.) If you want to do the problem set by hand, please submit it at the beginning of class.
- Start each problem on a separate page.
- If you submit the problem set on paper, make sure your name is on each sheet of paper (and legible).
- If you submit the problem set on paper, staple the pages together.

Remember, that when a question asks for an algorithm, you should:

- First, give an overview of your answer. Think of this as the executive summary.
- Second, describe your algorithm in English, giving pseudocode if helpful.
- Third, give an example showing how your algorithm works. Draw a picture.

You may then give a proof of correctness, or explanation, of why your algorithm is correct, an analysis of the running time, and/or an analysis of the approximation ratio, depending on what the question is asking for.

Advice. Start the problem set early—questions may take time to think about. Come talk to me about the questions. Talk to other students about the problems.

Collaboration Policy. The submitted solution must be your own unique work. You may discuss your high-level approach and strategy with others, but you must then: (i) destroy any notes; (ii) spend 30 minutes on facebook or some other non-technical activity; (iii) write up the solution on your own; (iv) list all your collaborators. Similarly, you may use the internet to learn basic material, but do not search for answers to the problem set questions. You may not use any solutions that you find elsewhere, e.g. on the internet. Any similarity to other students' submissions will be treated as cheating.

Review (***Not to be submitted.***)

Exercise 1. [Variance]

Recall that for a random variable X, the variance of X $(\mathrm{Var}[X])$ is defined as $\mathrm{E}\left[(X-\mu)^2\right] = \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2$, where $\mu = \mathrm{E}[X]$.

a. Let X be an indicator random variable that is 1 with probability p and 0 with probability $1-p$. Prove that $\mathrm{Var}[X] = p(1-p)$.

b. Let X be the number of heads when you flip a coin n times where each flip is heads with probability $1/2$. Calculate $\mathrm{Var}[X]$.

Exercise 2. [Chebychev]

Given a random variable X, prove that:

$$\Pr[|X-\mu| \geq t] \leq \frac{\mathrm{Var}[X]}{t^2}$$

This is known as Chebychev's Inequality. (Hint: use Markov's Inequality to bound $\Pr\left[(X-\mu)^2 \geq t^2\right]$.)

Exercises (***To be done, but may not be submitted.***)

Exercise 3. [Balls in Bins]

Assume you have m balls and n bins. Each ball is placed in a bin chosen uniformly at random. Let X_i be the number of balls in bin i.

a. What is $\mathrm{E}[X_i]$?

b. What is $\Pr[X_i = 0]$?

c. For $m = n$, show that the expected number of empty bins (after the process is complete) is approximately n/e.

d. For $m = 2n \log(n)$, show that the expected number of empty bins (after the process is complete) is < 1.

Exercise 4. [QuickSelect]

Consider the QuickSelect algorithm which returns the kth item in an (unsorted) array:

Algorithm 1: QuickSelect$(A, k, begin, end)$

```
$p \leftarrow \mathbf{Random(begin, end)}$
$i \leftarrow \mathbf{Partition(A, p)}$
if $k < i$ then return QuickSelect$(A, k, begin, i - 1)$.
if $k > i$ then return QuickSelect$(A, k - i, i + 1, end)$.
if $k = 1$ then return $A[i]$.
```

Prove that QuickSelect takes $O(n)$ expected time.

Standard Problems (to be submitted)

Problem 1. [Flipping Coins]

Imagine you flip a fair coin n times. Each time you flip the coin, it comes up heads with probability 1/2 and tails with probability 1/2. A *streak of length* k occurs when the coin comes up heads k times in a row at some point during the sequence. For example, consider the following sequence:

T T H H H T H T T T H H H H

In this sequence there is one streak of length 4, three streaks of length 3, and five streaks of length 2.

Problem 1.a. What is the expected number of streaks of length k?

Problem 1.b. Show that the probability of a streak of length $\log n + 2$ is small (e.g., $< 1/2$).

Problem 1.c. Show that, for sufficiently large n, the probability that there is no streak of length at least $\lfloor \log n - 2 \log \log n \rfloor$ is less than $1/n$.

Hint: break the sequence of flips up into disjoint blocks of $\lfloor \log n - 2 \log \log n \rfloor$ *consecutive flips, and use the fact that streaks in different blocks are independent.*

Problem 2. [Estimators]

Imagine you have a large array A of size n containing all 0's and 1's. (The array might represent a large population, and $A[i]$ might indicate whether the person likes laksa.) We want to estimate the number of $1's$ in the array. That is, let $T = \sum_{i=1}^{n} A[i]$ be the number of $1's$ in array A.

We solve this problem via sampling. We repeat the following procedure s times: choose a random index in the array and check whether it is a 0 or a 1. Let X_i be the value of the ith sample. Let $Y = (n/s)\sum_{i=1}^{s} X_i$. The value Y is an estimator for the sum T.

a. What is $\mathrm{E}[X_i]$?

b. What is $\mathrm{Var}[X_i]$?

c. What is $\mathrm{E}[Y]$?

d. What is $\mathrm{Var}[Y]$? (Hint: Show that $\mathrm{Var}[Y] = (n^2/s)\mathrm{Var}[X_1]$.)

From these results, do you expect that Y is a good estimator of T?

Problem 3. [Permutations]

For Chinese New Year, we all decide to play Secret Santa[1]. For a class of size n, we each put one present into a big pile (leading to a pile of n presents), and we each get a random present from the pile.

Problem 3.a. What is the expected number of people who get their own present (i.e., who get from the pile the same present they put into the pile)?

Problem 3.b. In fact, the assignment of presents to people is a random permutation. Let π be a random permutation of $[1, \ldots, n]$, uniformly chosen from all possible permutations of n. We say that $v_1, v_2, \ldots, v_k$ is a cycle of length k if $\pi(v_i) = v_{i+1}$ for all $i < k$, and $\pi(v_k) = v_1$. For example, consider the following permutation:

$$\sigma = \mathbf{3\ 5\ 1\ 2\ 4}$$

(That is, $\sigma(1) = 3$, $\sigma(2) = 5$, etc.) This permutation consists of two cycles: $(1, 3)$ and $(2, 5, 4)$. The first cycle has length 2 and the second cycle has length 3.

Fix a person j. How many cycles of length 1 contain person j? How many cycles of length 2 contain person j? How many cycles of length x contain person j?

Problem 3.c. Every permutation can be decomposed into cycles. What is the expected number of cycles in a randomly chosen permutation π?

Hint: Define the random variable S_i to be the size of the cycle containing i. For example, in the permutation σ above, $S_1 = 2$, $S_2 = 3$, $S_3 = 2$, $S_4 = 3$, $S_5 = 3$. Define the random variable Y_i to be $1/S_i$. Again, in the example above for permutation σ, $Y_1 = 1/2$, $Y_2 = 1/3$, $Y_3 = 1/2$, $Y_4 = 1/3$, $Y_5 = 1/3$.

Problem 3.d. (Optional.) For some $k > n/2$, what is the probability that the permutation contains a cycle of length k? What is the probability that the permutation contains no cycle of length $> n/2$? (Hint: notice that if a permutation has a cycle of length $k > n/2$, then it does *not* have a cycle of length $k' \neq k$ where $k' > n/2$.)

[1] Why can't Santa deliver presents for Chinese New Year too? He's had plenty of rest since Christmas.

Challenge Problem (Optional)

Problem 4. Imagine you have n bins and n balls. Consider the following game:

> Repeat the following until there are no bins left:
>
> – Place each ball into a bin chosen uniformly at random from the remaining set of bins.
>
> – Remove every bin that has at least one ball.
>
> – Collect all n balls and continue.

For example, after the first iteration, in expectation, there will be about n/e bins left. The process terminates when there are no bins left. Prove that, in expectation, the process terminates in $O(\log^* n)$ rounds, where $\log^* n$ is the iterated logarithm[2].

[2] Define $\log^* 1 = 0$ and $\log^* n = 1 + \log^*(\log n)$.