NYU TANDON SCHOOL OF ENGINEERING

New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 6763: Homework 1.
Due Tuesday, September 26th, 2023, 11:59pm ET.

Collaboration is allowed on this problem set, but solutions must be written-up individually. Please list collaborators for each problem separately, or write "No Collaborators" if you worked alone.

For just this first problem set, 10% extra credit will be given if solutions are typewritten (using LaTeX, Markdown, or some other mathematical formatting program).

Problem 1: Short(er) problems. (8 pts)

1. (4pts) Consider inserting m keys into a hash table of size $n = 5m^2$ using a uniformly random hash function. By the mark-and-recapture analysis from class, we know that the expected number of collisions in the table is $\frac{m\cdot(m-1)}{2\cdot 5m^2} < 1/10$. So, by Markov's inequality, with probability $> 9/10$, the table has no collisions (< 1 collisions). Thus, we can look up items from the table in worst-case $O(1)$ time.

   Give an alternative proof of the fact that we have no collisions with $> 9/10$ probability in a table of size cm^2 for some sufficiently large constant c. Specifically, to have no collisions, we must have the following events all happen in sequence: the second item inserted into the hash table doesn't collide with an existing item, the third item inserted doesn't collide with an existing item, …, the m^{th} item inserted doesn't collide with an existing item. Analyze the probability these events all happen. **Hint:** You might want to use the fact that $\frac{1}{2e} \leq (1-\frac{1}{n})^n \leq \frac{1}{e}$ for any positive integer $n \geq 2$.

2. (4pts) Consider a random walk on a d-dimensional grid. At each step, the walk chooses one of the d-dimensions uniformly at random, and takes a step in that direction – up with probability 1/2 and down with probability 1/2. Assume that the walk starts at the origin, takes n steps, and ends at position x, where x is a d dimensional vector with integer values. Show that $\mathbb{E}[\|x\|_2^2] = n$, where $\|x\|_2^2 = \sum_{i=1}^d x_i^2$ denotes the squared Euclidean norm of x.

   I find this fact surprising. If a tourist starts at Washington Square Park and wonders around Manhattan (i.e., takes a two-dimensions random walk), in expectation they don't get any more lost than if they restrict their wondering to just up and down 5th Avenue (i.e., a one dimensional random walk.)

Problem 2: Why does Count-Min work so well in practice? (12 pts)

We showed that Count-Min can estimate the frequency of any item in a stream of n items up to additive error $\frac{1}{m}n$ using $O(m)$ space. In practice it is often observed that this bound is pessimistic: the algorithm performs better than expected. In this problem, you will establish one reason why.

For any positive integer m, let $f_1, \ldots, f_m$ be the frequencies of the m most frequent items in our stream. Let $C = n - \sum_{i=1}^m f_i$. In general, we can have that $C \ll n$. For example, it has been observed that up to 95% of YouTube video views come from just 1% of videos. Prove that using $O(m)$ space, Count-Min actually returns an estimate $\tilde{f}$ to $f(v)$ for any item v satisfying:

$$f(v) \leq \tilde{f} \leq f(v) + \frac{1}{m}C$$

with 9/10 probability. This is strictly better than the $\frac{1}{m}n$ error bound shown in class.

Problem 3: Randomized methods for COVID-19 group testing. (12 pts)

One of the most important factors in controlling diseases like COVID-19 is testing. Before at-home kits became available, testing was expensive and slow. One way to make it cheaper was to test patients in *groups*. The biological samples from multiple patients (e.g., multiple nose swabs) are combined into a single test tube and tested for COVID-19 all at once. If the test comes back negative, we know everyone in the group is negative. If the test comes back positive, we do not know which patients in the group actually had COVID-19, so further testing would be necessary. There's a trade-off here, but it turns out that, overall, group testing can save on the total number of tests run.

1. (6pts) Consider the following deterministic "two-level" testing scheme. We divide a population of n individuals to be tested into C groups of the same size. We then test each of these groups. For any group that comes back positive, we retest all members of the group individually. Show that there is a choice for C such that, if k individuals in the population have COVID-19, we can find all of those individuals with $\leq 2\sqrt{nk}$ tests. You can assume k is known in advance (often it can be estimated accurately from the positive rate of prior tests). This is already an improvement on the naive n tests when $k < 25\% \cdot n$.

2. (6pts) We can use randomness to do better. Consider the following scheme: Collect $q = O(\log n)$ nose swabs from each individual (I know... not pleasant). Then, repeat the following process q times: randomly partition our set of n individuals into C groups, and test each group in aggregate. Once this process is complete, report that an individual "has COVID" if the group they were part of tested positive all q times. Report that an individual "is clear" if *any* of the groups they were part of tested negative. Show that for $C = O(k)$, with probability $9/10$, this scheme finds all truly positive patients and reports no false positives. Thus, we only require $O(k \log n)$ tests!

3. **Extra Credit – optional.** (3pts) Show that no scheme can use $o(k \log(n/k))$ tests and succeed with probability $> 2/3$. So, for small k, the approach above is essentially optimal up to constant factors!

Problem 4: Exploring Concentration Bounds. (8 pts)

Using combinatorics, write down and evaluate (e.g., run some code) an exact expression for the probability of flipping 60 or more heads when flipping a fair coin 100 times. Do the same for 600 or more heads in 1000 flips. Compare your results to the upper bound given by the Chernoff bound from Lecture 2:

$$\Pr[S > (1+\epsilon)\mu] \leq e^{-\frac{\epsilon^2 \mu}{2+\epsilon}}.$$

Also compare to this slightly different and less commonly used form of the bound from Wikipedia:

$$\Pr[S > (1+\epsilon)\mu] \leq \left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu}.$$

Also compute what each Chernoff Bound would give if you wanted to bound the probability of flipping more than 70 or 700 heads. Write a 1-2 sentence comment on how what your findings suggest about how "tight" the Chernoff bound is.

Problem 5: Try out mark-and-recapture! (12 pts)

1. (6pts) Prove the claim on Slide 32 of Lecture 1. Specifically, if we collection $O(\sqrt{n}/\epsilon)$ samples uniformly from a set of unknown size n, then we can return an estimation $\tilde{n}$ which, with probability $9/10$ satisfies:

$$(1-\epsilon)n \leq \tilde{n} \leq (1+\epsilon)n.$$

2. (6pts) Wikipedia provides a way to access a random article by following the link `https://en.wikipedia.org/wiki/Special:Random`. Implement the mark-and-recapture algorithm from class and use this link to evaluate the claim that English Wikipedia has 6.7 million unique articles. In my experiments, downloading 5000 articles took 30 minutes, so scanning all possible articles to check the claim would take ~ 25 days (if you don't get blacklisted for scrapping first). How long did your code take to run?

   Include your code, all relevant results and calculations, and a discussion of how accurate you think your estimate is. Do you think its accuracy will be impacted by the fact that the random article feature doesn't return *perfectly* random articles (see discussion here)?

   I used Python, but you can use any language. In Python, you can get a random url by running:

```
import requests
response = requests.get("https://en.wikipedia.org/wiki/Special:Random")
random_url = response.url
```