New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 6763: Homework 1.
Due Sunday, Jan. 25th, 2026, 11:59pm ET.

Problem 1: Collision Free Hashing.

Consider inserting m keys into a hash table of size $n = 5m^2$ using a uniformly random hash function.

1. Consider inserting m keys into a hash table of size $n = 5m^2$ using a uniformly random hash function. Leverage the mark-and-recapture analysis from class to show that the expected number of collisions in the hash table is $< 1/10$. Prove that there will be no collisions between inserted items with probability $> 9/10$. Thus, we can look up items from the table in worst-case $O(1)$ time.

2. 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 $n \geq 2$.

Problem 2: High-dimensional Random Walks.

Consider inserting m keys into a hash table of size $n = 5m^2$ using a uniformly random hash function.

1. 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. Prove 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 because the expected distance does not depend on the dimension d. I.e., if a tourist starts at Washington Square Park and randomly 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 3: Try out mark-and-recapture! (Optional)

This is a fun coding problem. You won't be tested on this, but I encourage you to try it out if you have time.

Wikipedia provides a way to access a random article by following the link https://en.wikipedia.org/wiki/Special:Random. For this problem, you will implement the mark-and-recapture algorithm from class and use this link to evaluate the claim that Wikipedia has ~ 7.1 million unique articles. In my experiments, downloading 5000 articles took a couple hours minutes, so scanning all possible articles to check the claim would take $\sim$ 100 days (if you don't get blacklisted for scrapping first). This is the exactly the sort of application where mark-and-recapture is useful!

1. Write and run your code. If you solve the problem, turn in the code, your best estimate for the number of articles on Wikipedia, and how many samples you used to compute the estimate.

   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
```

2. **Fun Theory Challenge.** You might notice that your estimate above seems to underestimate Wikipedia's claimed number of articles. A prior student in this class figured out that this is due to the fact that Wikipedia's random article generator does not return *truly uniform* random articles. As discussed here, Wikipedia assigns each article i a random id r_i, which we can model as a random real number in $[0, 1]$. Then, to pick a random article, a number is sampled uniformly from $[0, 1]$ article i is returned if that number lies in the range $[r_i, r_{i+1}]$. Since these intervals themselves are random, the probability distribution won't be *perfectly* uniform.

   Easy-ish. Prove that, when the interval lengths are not perfectly uniform, the mark-and-recapture method will systematically underestimate the number of articles. I.e., it will return an underestimate even as the number of samples $m \to \infty$.

   Harder. Prove that, if Wikipedia uses the scheme above, we expect that the mark-and-recapture method to underestimate the number of articles by *almost exactly a factor of two*. We could thus correct for this in our estimate.