Homework 7
Colorado CSCI 5454

Alex Book

October 26, 2021

People I studied with for this homework: Cole Sturza
Other external resources used: N/A

Problem 1

The balls are numbered $i = 1, \ldots, n$. Let

$$X_{abc} = \begin{cases} 1 & a,\ b, \text{ and } c \text{ go into the same bin} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} \mathbb{E}[X_{abc}] &= \sum_{k=1}^{\ell} Pr[a, b, \text{ and } c \text{ all land in bin } k] \\ &= \sum_{k=1}^{\ell} \frac{1}{\ell}\frac{1}{\ell}\frac{1}{\ell} \\ &= (\ell)(\frac{1}{\ell^3}) \\ &= \frac{1}{\ell^2} \end{aligned}$$

Now, we can calculate the expected number of 3-way collisions:

$$\begin{aligned}\mathbb{E}[\text{number of 3-way collisions}] &= \mathbb{E}\left[\sum_{\text{triplets } a,b,c} X_{a,b,c}\right] \\ &= \sum_{\text{triplets } a,b,c} \mathbb{E}[X_{a,b,c}] \\ &= \sum_{\text{triplets } a,b,c} \frac{1}{\ell^2} \\ &= \binom{n}{3}\frac{1}{\ell^2} \\ &= \Theta(\frac{n^3}{\ell^2})\end{aligned}$$

We conclude that with ℓ bins, we expect a 3-way collision after throwing just $n = \Theta(\ell^{\frac{2}{3}})$ balls.

Problem 2

Claim: If you hash all items in U into ℓ bins, $\exists$ at least one bin with at least ℓ items.

Proof:
Let X_j be the number of items in bin j.

$$\begin{aligned}\text{Average number of items per bin} &= \frac{1}{\ell}\sum_{j=1}^{\ell} X_j \\ &= \frac{|U|}{\ell} \\ &\geq \frac{\ell^2}{\ell} \\ &= \ell\end{aligned}$$

So, the average number of items per bin is ℓ. We can also make the following statement (that the maximum number of items in any bin must be greater than or equal to the average number of items per bin):

$$\max_j X_j \geq \tfrac{1}{\ell}\textstyle\sum_{j=1}^{\ell} X_j \geq \ell$$

Therefore, we can say with certainty that there exists at least one set of at least size ℓ that is all hashed into the same location/bin.

Problem 3

Part a

1.

A Bloom filter is appropriate for this task, as it does not allow false negatives. Additionally, it will be a very quick lookup process (no collisions, simply checking if the hashed locations are all equal to 1).

2.

A hash table is appropriate for this task, as it will guarantee that the answer is correct (a VIP user will be found in the table, while a non-VIP user will not), as well as fetching each VIP's special information (it's possible there will be collisions, but this is taken care of using chaining to keep each entry "separate").

3.

A Count-min sketch is appropriate for this task, as it will never undercount a user's contributions. Additionally, the lookup time and is faster and space used is less than those of a hash table.

Part b

1.

Let the acceptable false positive rate be .5%. As detailed in lecture notes on Bloom filters, we can set parameters k and m based on the desired false positive rate.

$$k = \log_2(\tfrac{1}{\delta}) = \log_2(\tfrac{1}{.005}) \approx 7.644$$
$$m = \tfrac{nk}{\ln(2)} = \tfrac{1{,}000{,}000 \cdot 7.644}{\ln(2)} \approx 11,027,960.893$$

So, we are left with roughly 8 hashing function and need an array of length 11,027,961. Since each array element is one bit, we can say that we use about 1.378 MB of storage.

2.

Let the initial ℓ (length of the array of hash locations) be 100. This seems to be a "happy medium" between minimizing collisions while also minimizing the amount of empty space (using 1000 bins would likely lead to many empty bins, while using 100 will likely lead to less empty bins). The average chain length in this case is 10 (since we are storing the data of 1000 VIPs and our hash table length is 10), so the average case for searching for a VIP's special information is searching through the 10 profiles that are stored at their hash location. As detailed here, the average Facebook user account is estimated to contain roughly 500 MB of data (note that this is a very rough estimate, but fun to consider for the sake of this problem). We're storing the data of each VIP user, so we can say that we use about 500 GB of storage.

3.

Let the desired count accuracy be within .5%, except with probability $e^{-10} \approx .00005$. As detailed in lecture notes on Count-min sketches, we can set parameters w and d based on the desired accuracy parameter ϵ and error-probability bound δ.

$$w = \lceil \tfrac{e}{\epsilon} \rceil = \lceil \tfrac{e}{.005} \rceil = \lceil 200e \rceil$$
$$d = \lceil \ln \tfrac{1}{\delta} \rceil = \lceil \ln(\tfrac{1}{e^{-10}}) \rceil = 10$$

So, we are left using a two-dimensional array of size $\lceil 200e \rceil \times 10$ for a total array size of 5440. Assuming we use 32-bit integers, our array has a total of 174,080 bits, so we can say that we use about 21.76 kB of storage.