CS 170 Section 12

Hashing, Streaming

Owen Jow

April 18, 2018

University of California, Berkeley

Table of Contents

Hashing Intro

Exposition

- Have: a bunch of data items from a large universe U
- Want: storage scheme allowing for $O(1)$ lookup, insertion, etc.
- Solution: **chained hash table**
- Need: a hash function that distributes items evenly into buckets

Chained Hash Table

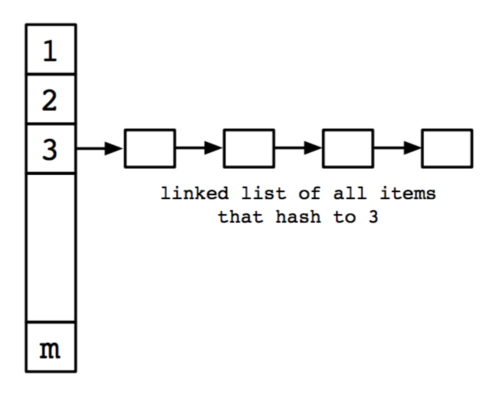

Figure 1: hash table. An item $x \in U$ hashes to a bucket $\{1, ..., m\}$ through some hash function $h(x)$, which takes an item and outputs a bucket index.

Hash Function

$$h : U \mapsto \{1, ..., m\}$$

- Observation: *no hash function performs well for all possible datasets*
- So choose one randomly from a universal family $\mathcal{H}$ of hash functions
- Most $h \in \mathcal{H}$ should perform well for any given dataset
- Why would this be good?

Universal Family

Universal family $\mathcal{H}$ of hash functions:

- For $y \neq z$ and a hash function h selected randomly from $\mathcal{H}$,

$$P(h(y) = h(z)) <= \frac{1}{m}$$

i.e. $\leq \frac{|\mathcal{H}|}{m}$ of all $h \in \mathcal{H}$ map y and z to the same bucket

Streaming Intro

Exposition

- Have: a sequence of incoming data $x_1, ..., x_n$
- Want: to compute features of the sequence without storing it all
  - e.g. heavy hitters, # distinct values, sum of squares of frequencies F_2
  - $\approx O(\log n)$ bits of memory?
- *Use randomized algorithms that provide approximate solutions*

CS 70 Strikes Again

- In dealing with randomized algorithms, we'll want to perform probabilistic analysis. Here's a bound that will come in handy...

Prove the Markov inequality

$$P(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

Hint: start with the definition of expectation

$$\mathbb{E}[X] = \sum_x xP(X = x)$$