CS 365, Lecture 12
Foundations of Data Science
Boston University

Charalampos E. Tsourakakis

March 15th, 2022

Dictionary problem

Universe $U = [u] = \{0, \ldots, u-1\}$

Set $S \subseteq U$, $|S| = n$, $|S| \ll U$

Goal: design a data structure that supports efficiently the following operations.

- $\textsc{Make}()$: Initializes an empty dictionary
- $\textsc{Insert}(\textsc{x})$: Add element x in S
- $\textsc{Lookup}(\textsc{x})$: Does x appear in S
- $\textsc{Delete}(\textsc{x})$: Removes x from S, if present

Questions:

- Why not a linked list?
- Why not an array over U?

Python dictionary

```
# empty dict
d ={}

#insert
d["Greece"] = "Athens"
d["France"] = "Paris"
d["Spain"] = "Madrid"
d["Italy"] = "Rome"

#lookup
print(d["Italy"])

#delete
del d["Spain"]
print(d["Spain"])#KeyError:'Spain'
```

Hashing

- **Basic idea:** Work with an array of size $m = O(|S|)$ rather than of size $O(|U|)$!

- **Hash function:** $h : [u] \to [m]$

- **Hash table:** Array. We place $x \in S$ at position $h(x)$.

- **Collision:** $x \neq y \in U$ get mapped to $h(x) = h(y)$.

1. How do we choose h?
2. How do we resolve conflicts?

Balls and Bins: k-wise independence

Consider the load of some bin.

$$\sum_{S \subseteq [n], |S|=k} \frac{1}{r^k} \leq \left(\frac{en}{k}\right)^k r^{-k} = \left(\frac{en}{rk}\right)^k$$

- No need for full randomness, but randomness over all subsets of k hash values.
- This naturally leads as to ***k*-wise independence**

Balls and Bins: k-wise independence

Definition: RVs $X_1, \ldots, X_n$ are k-wise independent iff for any set of indices $i_1, \ldots, i_k$, RVs $X_{i_1}, \ldots, X_{i_k}$ are independent.

Definition: A set of hash function $\mathcal{H}$ is a k-wise independent family iff the random variables $h(0), \ldots, h(u-1)$ are k-wise independent when $h \in \mathcal{H}$ is drawn uniformly at random.

Example 1: The set $\mathcal{H}$ of all functions from $[u]$ to $[m]$ is k-wise independent for all k.

 # Bits: $u \log m$ (u is enormous!)

Construction

We can construct a 2-wise independent family as follows.

- p is prime
- a, b chosen uar from $[p]$
- The hash of x is

$$h(x) = ax + b \mod p,$$

How many bits do we need now?

Generalization: Polynomials with random coefficients

- Choose k random numbers modulo p (p large prime), say $a_0, \ldots, a_{k-1}$.
- $h(x) = \sum_{i=0}^{k-1} a_i x^i \mod p$