COL 702 Lecture 8 Aug 16

Dictionary : search, insert, delete

Balanced BST are efficient dictionaries : $O(\log n)$ for each operation

set of elements S Dictionary (S)

$|S| = n$ S is totally ordered

Question : Can we do better?

Define a universal set $\mathcal{U}$ of keys

$$\mathcal{U} = \{0, 1, 2, \cdots N-1\}$$

hash function $h : \mathcal{U} \to T : \{0, 1, .. m-1\}$

$$|\mathcal{U}| \gg |T|$$

Any instance of building a dictionary is

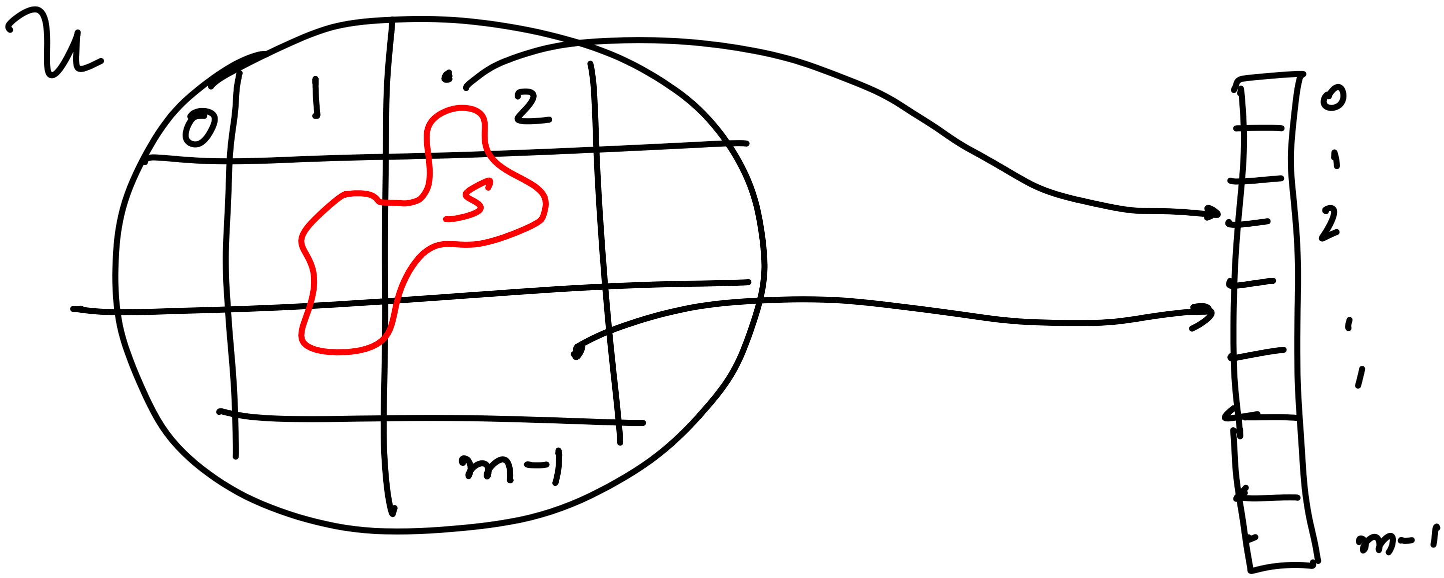
Given a subset $S \subset \mathcal{U}$,

$h : S \to T$ $|S| \leq |T|$

$|S| = n$ otherwise collision

$\forall\ x, y \in \mathcal{U}$ x, y collide if
$h(x) = h(y)$

A hash function is 'good' if there are no collisions within S



A good hashing scheme should work for all subsets S

$h(x) = x \bmod m$

What happens for a random subset S

→ Each of the n elements of S is chosen uniformly at random from $\mathcal{U}$

→ (Strictly speaking $\binom{N}{n}$ one of them is chosen at random)

What is performance of the hash function mod m?

What is the expected # of elements of S that will fall into location 0?

Recall that when elements collide in the hash table $\rightarrow$ build a linked list

$\rightarrow$ probing sequence

By using the linked list, the performance of hashing is proportional to the length of the list

$S = \{x_1, x_2 \ldots x_n\}$

$\text{Prob}(x_1 \in T(0)) = \frac{1}{m}$

Let Z_i be a r.v. $= \begin{cases} 1 & \text{if } x_i \in T(0) \\ 0 & \text{otherwise} \end{cases}$

$\text{Prob}[Z_i = 1] = \frac{1}{m}$

elements in $T(0) = Z_1 + Z_2 + \cdots Z_n = Z$

$\Rightarrow E[Z] = E[\sum_i Z_i] = \sum_i E[Z_i] = n \cdot \frac{1}{m}$

$\left(E[Z_i] = \frac{1}{m} \right)$ (Read Ch 2 old notes)

If $n = m$ $E[Z] = 1$

$\Rightarrow$ $\text{Prob}(Z \geq 2) \leq \frac{1}{2}$ from Markov

We would like to analyze the performance over a sequence of operations

$O_1, O_2, O_3 \ldots O_t$ $O_i \in \{$search, insert, delete$\}$

$$E[O_i] = \frac{n}{m}$$

$$E[O_1 + O_2 + O_3 \cdots O_t] = \sum E[O_i]$$
$$= t \cdot \frac{n}{m}$$

What if S is worst case / arbitrary?

Assume that we have a family of hash functions $\mathcal{H}$

Basic hope: Given an arbitrary S there is at least one good $h \in \mathcal{H}$ for S

Idea: Pick a random $h \in H$

A family of hash function H is called c-universal if

$$\forall\, x, y \in \mathcal{U} \quad \sum_{h \in H} \delta_h(x,y) \leq c\,\frac{|H|}{m}$$

where c is some constant

$$\delta_h(x,y) = \begin{cases} 1 & \text{if } h(x) = h(y) \\ 0 & \text{otherwise} \end{cases}$$

collision function for h

Therefore if we pick a hash function at random, the probability that any given pair x, y will collide is $\frac{c}{m}$

What is the expected # collisions in location 0 of the table for a random hash function? (for an arbitrary S)

Consider a element x of S

$$\frac{1}{|H|} \sum_{h \in H} \sum_{\substack{y \in S \\ y \neq x}} \delta_h(x,y)$$: expected # collision for a fixed element $x \in S$

$$= \sum_{y \in S} \underbrace{\frac{1}{|H|} \sum_{h \in H} \delta_h(x,y)}_{\text{using defn of } c\text{-universal}}$$

$$= \sum_{y \in S} \frac{c}{m} \leq \frac{c \cdot n}{m}$$

EXISTENCE OF UNIVERSAL family ?