Hash Tables

Skip List

Have $2\log n$ levels to store n items.

Each level is a sorted linked list

When an item is inserted, flip a coin to determine whether it gets upgraded, if so, repeat.

Expected Search Cost?

Start

too far

H → 28 → 72

too far

H → 14 → 28 → 50 → 72

H → 14 → 23 → 28 → 34 → 42 → 50 → 66 → 72

Found it!

14				34
14		23		34
14	18	23	28	34

Expected Search Cost?

Start

H → 28 → 72

H → 14 → 28 → 50 → 72

H → 14 → 23 → 28 → 34 → 42 → 50 → 66 → 72

How many rightward steps do I make at level i?

Call this R_i. Cost of a query: $2\log n + \sum_i R_i$

Expected Search Cost?

Start ↘

H → 28 → 72

H → 14 → 28 → 50 → 72

H → 14 → 23 → 28 → 34 → 42 → 50 → 66 → 72

How many rightward steps do I make at level i?

Call this R_i. Cost of a query: $2\log n + \sum_i R_i$

Expected Search Cost

Start

H → 28 → 72

H → 14 → 28 → 50 → 72

H → 14 → 23 → 28 → 34 → 42 → 50 → 66 → 72

Starting from any position how many steps backwards before I find an upgraded item?

$\frac{1}{2}$ 1 step $\quad$ $\frac{1}{4}$ 2 steps $\quad$ $\frac{1}{8}$ 3 steps $\quad$ $\frac{1}{16}$ 4 steps.

Expected Search Cost

Starting from any position how many steps backwards before I find an upgraded item?

$\frac{1}{2}$ 1 step $\quad \frac{1}{4}$ 2 steps $\quad \frac{1}{8}$ 3 steps $\quad \frac{1}{16}$ 4 steps.

$$E\{R_i\} = \sum_j p(j)\, j = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16}$$

$$= \sum_{i=1}^{k} \frac{i}{2^i} \leq \sum_{i=1}^{\infty} \frac{i}{2^i} \leq 2.$$

Expected Query Cost

$$2\log(n) + \sum_i E[R_i] \leq 2\log(n) + 2\log n \cdot 2$$

$$= O(\log(n))$$

Hash Tables

Suppose I need a dictionary over the integers between 1 and n, where n is small, can use an array

1	2	3	4	5	6	7	8	9	10	11	12	13	....
		x			x		x			x			

What is a hash function?

Have a universe: e.g. integers between 0 and 2^{32}

strings of bytes

points in $\mathbb{R}^3$

A hash function is a randomized function

$$h : \mathcal{U} \to \{0, 1, \ldots, m\}$$

for some m.

Hash Table

Pick hash function h.

1 2 3 4 5 6 7 8 9 10 11 12 13 . . . . .

x x x

h("Alex") % 128 = 8

h("Jennifer") % 128 = 1

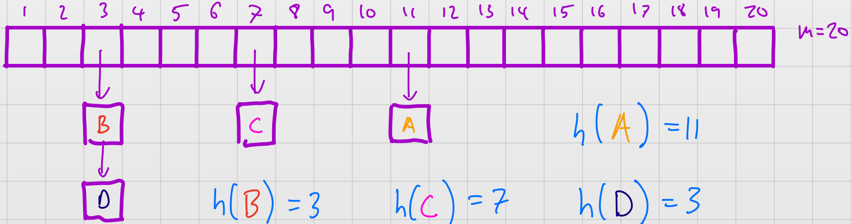
h("Toby") % 128 = 13

h("Raj") % 128 = 8

"Alex" and "Raj" are a <u>hash collision</u>

Chaining Hash Table

Hash each item to a bucket and store each bucket as a linked list

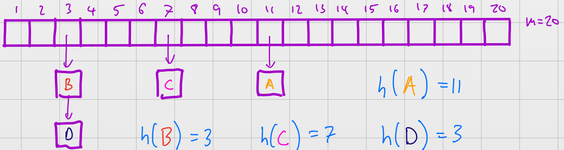


$h(A) = 11$

$h(B) = 3$ $h(C) = 7$ $h(D) = 3$

Chaining Hash Table

Hash each item to a bucket and store each bucket as a linked list



$h(A) = 11$

$h(B) = 3$ $h(C) = 7$ $h(D) = 3$

Cost of insertion:
$O(1)$

Cost of a query:
$O(\text{the length of the list})$

Expected Cost of a Query

Let C_t be the number of items hashing to t.

Let $x_1, \dots, x_n$ be the items in the hash table.

Let $C_{i,t} = \begin{cases} 1 & \text{if } h(x_i) = t \\ 0 & \text{otherwise} \end{cases}$ Then $C_t = \sum_{i=1}^{n} C_{i,t}$

$Pr(C_{i,t} = 1) = \frac{1}{m}$ $E\{C_{i,t}\} = 1 \cdot Pr(C_{i,t} = 1) + 0 \cdot Pr(C_{i,t} = 0)$

$= Pr(C_{i,t} = 1)$

$E(C_t) = \sum_{i=1}^{n} E\{C_{i,t}\} = n/m.$