COP3502 4/6/2022

1. Hash Tables
2. Quick Quiz Tips
3. CS Testing App

Balanced Binary Search Trees

Insert, delete $O(\log n)$

store "in order" (easy to retrieve next or prev item to a given item)

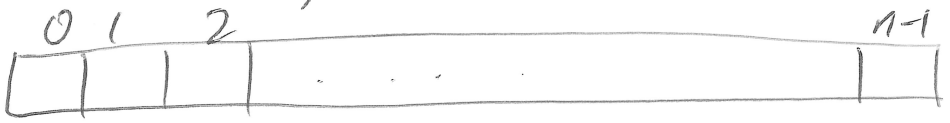
Can we do better?

Hash Table

insert, delete expected $O(1)$ time

UNORDERED

Create an array of size n (prime number)



magical thing called a "hash function"

Input: arbitrary sized inputs (usually strings or binary strings)

Output: a fixed integer in between 0 and n-1.

Many-to-one

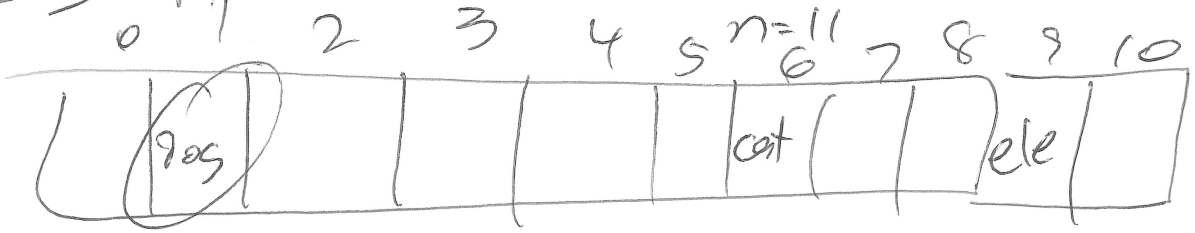
To be a good hash function

1) each output value should be equally likely

2) small input changes should result in larger changes in the output.

3) Given a hash value, it should be difficult to compute an input that creates that hash value.

⇒ RANDOM W/O PATTERNS



4) f should be fast to compute!

f("cat") = 6
f("dog") = 1
f("elephant") = 9
f("rabbit") = 1

↳ THIS IS A COLLISION

HOW TO DEAL?

1) DON'T (REPLACE ITEM)

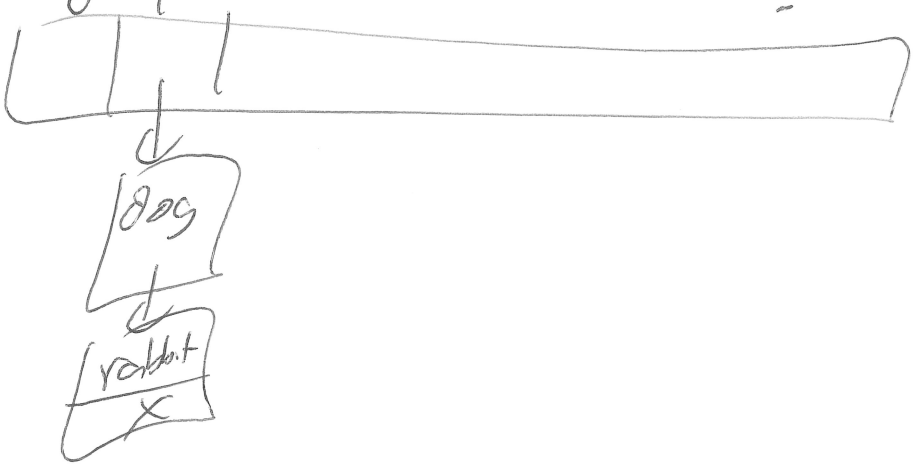
2) Linear Probing - if hash value is h, and slot h is filled, look in slots h+1, h+2, h+3, etc. until we find an opening

(h+i) % n

3) Quadratic Probing - if h is filled go to h+1, h+4, h+9, h+16, etc. $(h+i^2) \% n$

4) Linear Chaining Hashing

② each index Linked List!



Quadratic Probing

let n = prime # large (odd)

prove 1st $\frac{n-1}{2}$ slots are unique.

$(h+i^2) \bmod n$, $(h+j^2) \bmod n$

Proof by contradiction

assume $\exists\ i, j$ $0 \le i < j \le \frac{n-1}{2}$ s.t.

$$h + i^2 \equiv h + j^2 \pmod n$$

$$i^2 \equiv j^2 \pmod n$$

$$i^2 - j^2 \equiv 0 \pmod n$$

$$(i+j)(i-j) \equiv 0 \pmod n$$

Since n is prime $\Rightarrow$ $n \mid (i+j)$ or $n \mid (i-j)$

$i + j \ge 1$

$i + j < n-1$

false

$i \neq j$

can't happen

Use quadratic probing as long as
(a) table size is prime
(b) keep less than 1/2 filled

Write linked list code
array of linked list.

Reasonable hash func for a string

```
int hash = 0;
for (int i = 0; i < n; i++)
    hash = ((128*hash) + s[i]) % tablesize;
```