Hashing #2

CptS 223 - Fall 2017 - Aaron Crandall

Today’s Agenda

- Announcements
- Thing of the day
- Going over Midterm
- Hashing

Announcements

- I’m not fully done grading the midterm and getting everything uploaded to Blackboard.
- Next assignment will be on hashing - lovely, lovely, lovely hashing

/r/NotTheOnion:
A software engineer is detained for several hours by U.S. Customs — and given a test to prove he’s an engineer

http://www.cnbc.com/2017/02/28/software-engineer-detained-given-test-to-prove-hes-engineer.html

FTA: the officer presented him with a piece of paper and a pen and told him to answer the following questions:
"Write a function to check if a Binary Search Tree is balanced."

"What is an abstract class, and why do you need it?"

Going over midterm

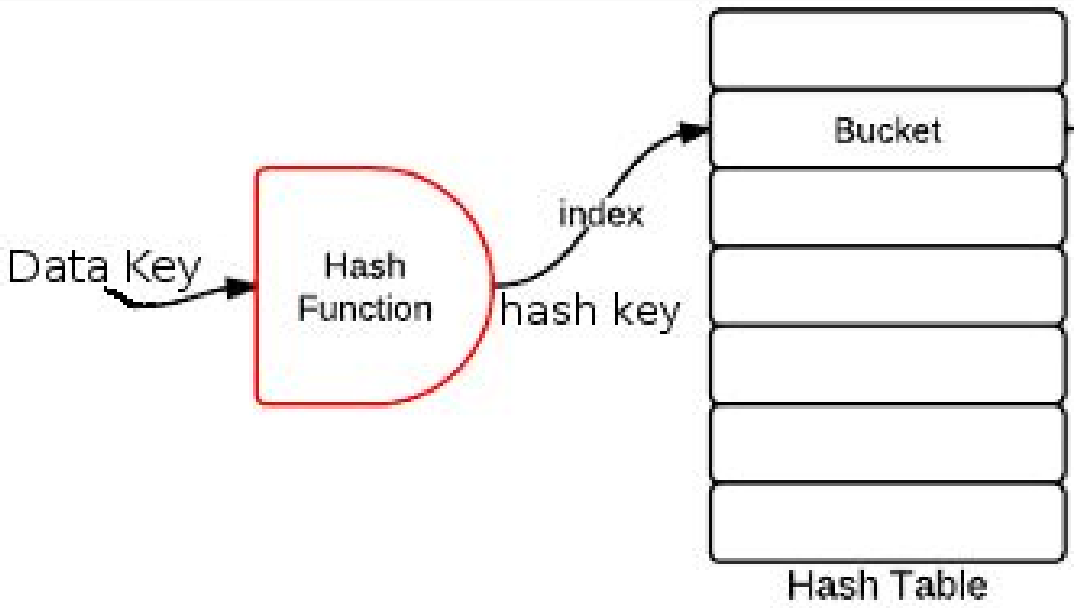
- Explaining the algorithm analysis will be the best part of the floor show.

Continuing Hashing!

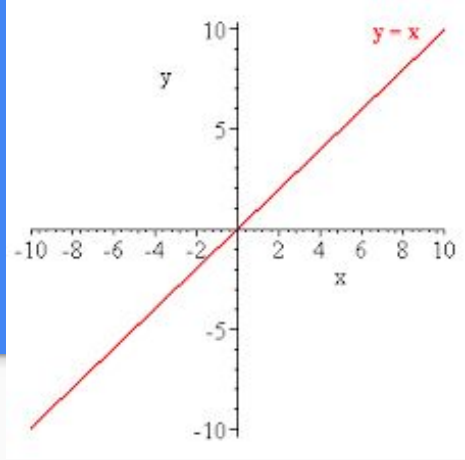
1) Hash table
    a) Normally a vector (array) of elements and indexed by hashed key
    b) Vector is of TableSize size
2) Hash Function
    a) Takes a key and returns the index in the hash table to place the record
    b) Ideally puts elements uniformly throughout the hash table
3) Collision resolution
    a) What do you do if two elements want the same index? - Normally inf keys, finite indexes

Doing an insert

- Take data's key
  - Integer, string, or other
- Run through hash function
  - Returns int for that key, hashkey
- Insert data into table
  - table[hashkey] = data;



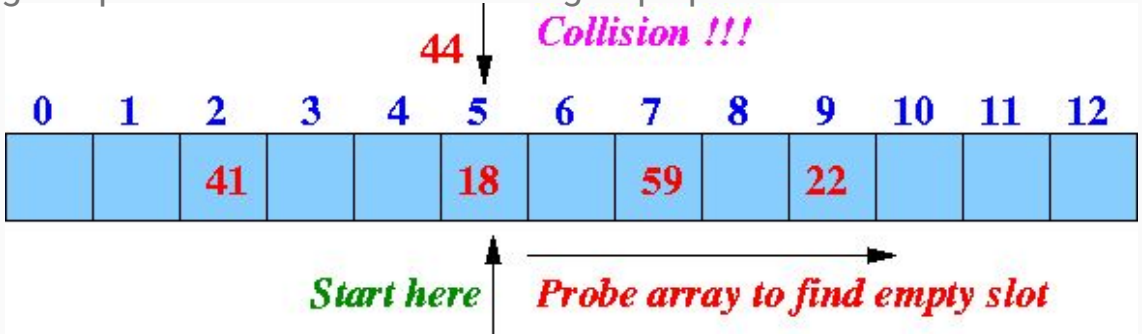
Doing an insert collision resolution: linear probing



- Take data's key
  - Integer, string, or other
- Run through hash function
  - Returns int for that key, hashkey This is a function: h(x)
- Try to Insert data into table
  - table[hashkey] = data;
- If collision (bucket already full) - then probe by adding probe function: f(i)
  - $h_i(x)$ = (hash(x) + f(i)) % TableSize, with f(0) = 0 -> while still collision, do i++ and repeat
  - Resolution strategy is f(i)
  - For linear probing: f(i) = i

Linear probing details

- Because the table holds the data (unlike chaining), table must be bigger
  - λ <= 0.5 -- Recall: chaining used λ ~= 1.0 λ == N / TableSize
- With linear probing, the f(i) algorithm is formally:
  - Collision resolution strategy: f(i) = i
  - This works as long as the table is big enough (see λ <= 0.5)
  - Primary clustering is a problem since numbers will group up in the table



Linear Probing Example: [89, 18, 49, 58, 69]

	Empty Table	After 89	After 18	After 49	After 58	After 69
0				49	49	49
1					58	58
2						69
3						
4						
5						
6						
7						
8			18	18	18	18
9		89	89	89	89	89

*note cluster!

Insert source code?

Go look at the source!

Need to look up find & hashString

Show in terminal (way better than in slides!)

Get, Delete, Contains, etc

- To reference anything in the hash table, you always need to pass the key through the hash function to get the hashkey!
- From there, it’s just an array (or vector, etc) like normal