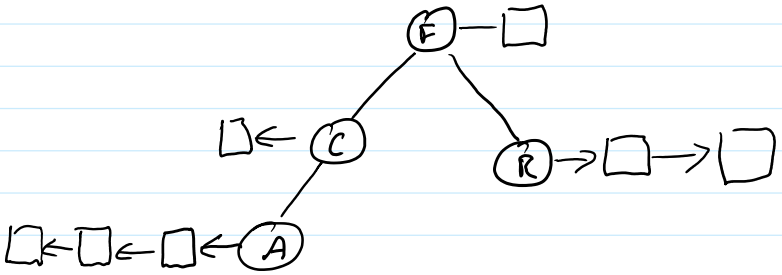
X

Monday, March 4, 2019 12:36 PM

## Overview:

- super trees!




Today:

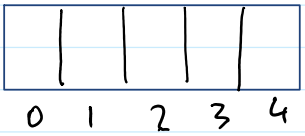
- review Kahoot Quiz from Friday
- intro to hash tables

## Hash Tables:

- hash table is a data structure that uses mapping to assign a "record" to a unique index of an array.
- Say, want to store the name and SID of a group of students

key: name

value: SID




hashFunction("Anna") ⇒ generates ⇒ 2 (e.g. unique index)

hashFunction("Jamie") ⇒ 0

"Bryan" ⇒ 1

### Two components:

1. array for storing the records

2. hash function for generating the "unique" code from a key.

↳ hash function is repeatable

hash("Billy") ⇒ 7

will always get a 7

↳ unique code is used as array index

Hash Functions: hashing a key that is string type

One of the simplest:

- sum values in a string
- then mod(%) by array length (table size)

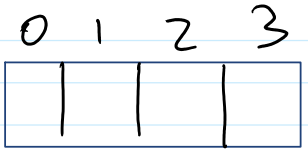
```
hashSum ( key, keyLength, tableSize )
    sum = 0
    for i = 0 to keyLength - 1
        sum = sum + key[i]

    return ( sum % tableSize )
```

E.G.

key = "Anna"

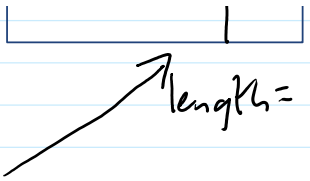
keyLength = 4

e.g. tableSize = 4:




↑ 0, ..., 4

e.g. tableSize = 4;



length = 4

Anna

A = 41
n = 110
n = 110
a = 97

358

358 % 4 = 2

if tableSize is 52

358 % 52 = 3

## Store Records in a hash table

1) Calculate index value (hash the key)
2) Write the data to hash table at the index

## Retrieve Records

1) use search key to calculate index
2) Read from the index

What is the cost of store? = O(1)

retrieve? O(1) or O(n)

## Collision

hashSum("Go Cat Go.") => 754 O(log n)

hashSum("Go Dog, Go") => 754

Each array location can only store a single record, but now we have 2 records trying to occupy one location.

$h(K_1) = h(K_2)$ $K_1 != K_2$