#27: Hashing Analysis
October 29, 2018 · *Wade Fagen-Ulmschneider*

Running Time Observations:
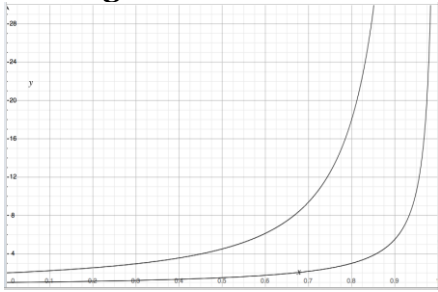
Linear Probing:
Successful: **½(1 + 1/(1-α))**
Unsuccessful: **½(1 + 1/(1-α))²**
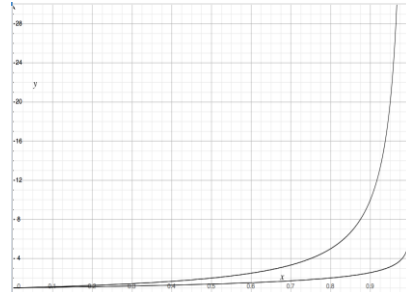
Double Hashing:
Successful: **1/α * ln(1/(1-α))**
Unsuccessful: **1/(1-α)**

ReHashing:
What happens when the array fills?

...or a better question:

Algorithm:

Which collision resolution strategy is better?
• Big Records:

• Structure Speed:

What structure do hash tables replace?

What constraint exists on hashing that doesn't exist with BSTs?

Why talk about BSTs at all?

Analysis of Dictionary-based Data Structures

	Hash Table		AVL	List
	SUHA	Worst Case		
Find				
Insert				
Storage Space				

Data Structures in std library:
- std::map

- std::unordered_map

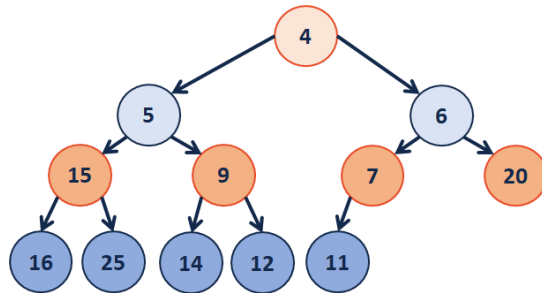## A Secret, Mystery Data Structure:

ADT:
insert

remove

isEmpty

Implementation of _____

insert	removeMin	Implementation
O(n)	O(n)	Unsorted Array
O(1)	O(n)	Unsorted List
O(lg(n))	O(1)	Sorted Array
O(lg(n))	O(1)	Sorted List

Q1: What errors exist in this table? (Fix them!)
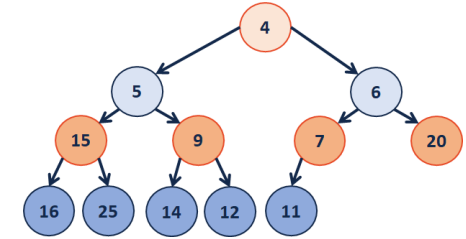
Q2: Which algorithm would we use?

A New Tree-like Structure:

A complete binary tree T is a min-heap if:

-
-

Implementing a (min)Heap as an Array

| 4 | 5 | 6 | 15 | 9 | 7 | 20 | 16 | 25 | 14 | 12 | 11 | | | |

Operations:

leftChild(index) :=

rightChild(index) :=

parent(index) :=

Insert:

| – | 4 | 5 | 6 | 15 | 9 | 7 | 20 | 16 | 25 | 14 | 12 | 11 | | | |