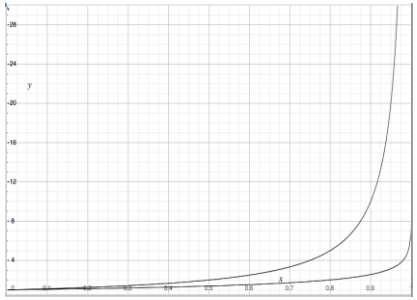
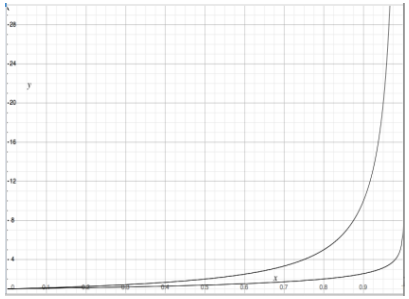
CS 225

#28: Hashing, Part 3
November 1, 2017

Running Time Observations:



Linear Probing:
Successful: $\frac{1}{2}(1 + 1/(1-\alpha))$
Unsuccessful: $\frac{1}{2}(1 + 1/(1-\alpha))^2$



Double Hashing:
Successful: $1/\alpha * \ln(1/(1-\alpha))$
Unsuccessful: $1/(1-\alpha)$

ReHashing:
What happens when the array fills?

Better question:

Algorithm:

	Hash Table		AVL	List
	Amortized	Worst Case		
Find				
Insert				
Storage Space				

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?

A Secret, Mystery Data Structure:

ADT:
insert

remove

isEmpty

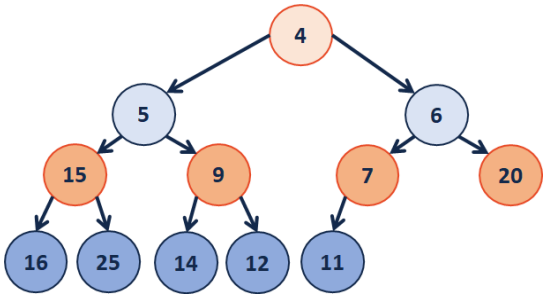
Priority Queue Implementations

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

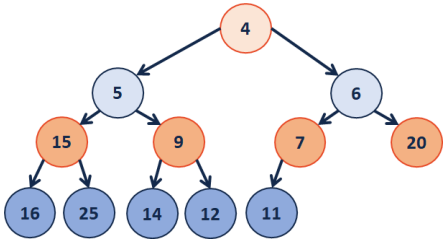
...what errors exist in this table?

Which algorithm would we use?

A New Tree-like Structure:



Implementing a (min)Heap as an Array



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

CS 225 – Things To Be Doing:

1. Register for CS 225's Final Exam!
2. Exam #8 (programming, MP4-like and AVL) onging
3. MP5 due Monday, Nov. 6
4. lab_heaps due Sunday, Nov. 5
5. Daily POTDs