

# CS130A S19 Final



Coakley

# Asymptotic Runtime

Know all the worst-case bounds (see practice final)

Know average case bounds where they truly differ (e.g. QuickSort)

Know the new tricky one: kDTree

# Recurrence Relation

Know the Master Method

Be able to produce a recurrence relation or compute runtime from a function pseudocode

# Hash Tables

When *isn't* it  $O(1)$ ?

What are good/bad properties of hash functions? Remember the discussion of naming you with numbers 0-39 and using the identity function as a hash function for a dense data set.

Birthday paradox: what is the probability of *at least 2* keys in a set of  $N$  keys colliding?

Know when to use a bloom filter and not a hash table.

# Heaps

What is the heap ordering property? Can you heapify?

$K^{n-1} = \sum_{i=0}^{n-1} (K-1)K^i$  ← the  $K-1$  out front is important, people have missed this before

Can you merge 2 leftist heaps? How fast?

# Search Trees

What properties do binary search trees have? What is the worst case (unbalanced, balanced)?

What are the properties of a valid AVL Tree?

What are the properties of a valid Red-Black Tree?

# Amortized Analysis

There are 3 standard questions:

The binary counter with some cost function

The multipop stack

Doubling some kind of array/table

If you know these, you are probably good

# Graphs

Topological Sorting

Mechanical execution of Prim's vs Kruskal's (what order would the edges be added?)

Apply shortest path algorithms



# B Trees

Use those formulas from wikipedia

Understand the relationships between minimum number of keys and maximum branching factor