Course Review

Graphs

Graph Representations
Adjacency List
Adjacency Matrix

Given a small graph, you should be able to illustrate these

Choose which is appropriate for an application
Graph Algs

Single Source Shortest Path Algs

Common use:

- Vertex $\leftrightarrow$ state in world
- edge $\leftrightarrow$ action that moves you between states
  - directed vs undirected
  - weighted vs unweighted
See when you can model a problem as a shortest path problem.

Unweighted - BFS starting at source (if there's a goal you can stop when you discover a path).

Weighted -

Fibonacci Dijkstra's alg - No negative weight edges are allowed.
Minimum Spanning Tree
Prim's & Kruskal's Alg

Depth First Search

Topological sort

In-place DFS & use in garbage collection (Mark & Sweep)
Spatial Collections

Kd-tree & quad tree

Digitized Ordered Collection

trie, compact Trie, compressed Trie

+ at high-level understand
adv & disadv to Ordered Collection
data structures
Ordered Collection

Binary Search Tree

Balanced Search Tree (Rotation)

Red-Black Tree properties & insertion

I won't ask about deletion

B-trees (+ brief intro to B^+ tree)

I won't ask about deletion (except in really

easy case like in homework)

only need to merge (reverse split)
Priority Queue
Binary Heap

Focus is up to here

Covered on Midterm

Hashing-based data structures
Open Addressing & Separate Chaining

† Also Direct Addressing
Sorting Algs

radix sort & counting sort
quick sort
merge sort
insertion

Know how these work, pros & cons,
which is the best choice
Adversary Lower Bound

Designing a strategy for an adversary to answer questions chosen by an algorithm to show how many questions any alg must ask so adv. only has one input left consistent with all past answers.
Asymptotic notation

Divide & conquer algo's