

CSCI 230 – Data Structures and Algorithms

Fall 2019 Final Study Guide

Note: The final exam is cumulative. This study guide covers all material after the midterm. For a study guide of the first half of semester material, please refer to the midterm study guide.

1. Recursion

- a. Transform recursive code to iterations (for or while loop)
- b. Solve a given problem recursively
- c. Calculate the best, worst, average case upper bound, lower bound, and runtime for a recursive algorithm.

2. Binary Trees

- a. Terminology – complete vs full trees, parents, siblings, path, level, height.
- b. Find the minimum/maximum number of internal nodes/leaves for a specific tree.
- c. Find the height of a tree.
- d. Find the level of a node.
- e. Pre/Post/In/Level order traversal for a given tree.
- f. Tree node implementation.
- g. Traversal implementation.

3. Binary Search Trees

- a. BST implementation.
- b. Add, remove node from given BST.
- c. Add a member function to a BST (findMax/Min, etc.)

4. Heaps & Priority Queues

- a. Array implementation of complete Binary Tree
- b. Heap implementation: buildHeap, insert, delete.
- c. Implement priority queues with a heap
- d. Decode a Huffman coding tree

5. Sorting Algorithms

- a. Algorithm implementation: insertion, bubble, selection, Shell, mergesort, quicksort, heapsort, binsort, radix sort.
- b. Calculate the best, worst, average case upper bound, lower bound, and runtime for a sorting algorithm – swaps and comparisons.

6. Hashing

- a. Hash and Probe functions
- b. Open vs closed hashing
- c. Insert/delete from hash table
- d. Rehash a table
- e. Calculate the load factor

7. Graphs (up to and including ch. 14.5)

- a. Terminology
- b. DFS – implementation, application of traversal
- c. BFS – implementation, application of traversal

- d. Topological sort
- e. Shortest path Dijkstra algorithm – implementation and application