

## **Final Exam Information**

**Date: 4/25/2024**

**Time: 1 PM – 3:50 PM**

**Room: CB1-121**

**The questions will vary from short answer, to tracing, to some coding.**

**Since there's a lot to remember algorithm wise, I'll allow you to use three 8.5"x11" sheets of notes, front and back.**

**Total Number of Points: 125**

**All Topics Possible**

**More Emphasis on Topics since Exam 2: Divide and Conquer and Dynamic Programming**

## **Final Exam Review Outline**

### **I. Use of Java Data Structures, Java Features**

- a. ArrayList, LinkedList, ArrayDeque**
- b. TreeSet, TreeMap**
- c. HashSet, HashMap**
- d. PriorityQueue**
- e. Custom Sorting**

### **II. Backtracking**

### **III. Data Structures**

- a. Disjoint Sets**
- b. 2-4 Trees**
- c. Red-Black Trees**
- d. Skip Lists**

### **IV. Algorithm Analysis**

- a. Accurate  $O$ ,  $\Omega$ ,  $\Theta$  definitions**
- b. Master Theorem**
- c. Expectation Definition**
- d. Binomial Theorem**
- e. Binary Search Average Case Run Time**
- f. Quick Sort Average Case Run Time**
- g. Quick Select Average Case Run Time**

### **IV. Sorting**

- a. Lower Bound for Adjacent Element Swap Sorts**
- b. Lower Bound for Comparison Sorts**
- c. Counting Sort**
- d. Radix Sort**

## **V. Greedy Algorithms**

- a. Fractional Knapsack**
- b. Single Room Scheduling**
- c. Multiple Room Scheduling**
- d. Change with certain denominations**
- e. Planting Trees and Similar Problems**
- f. Containers**
- g. Huffman Coding**

## **VI. Unweighted Graphs**

- a. Definition & Different Types**
- b. Depth First Search**
- c. Breadth First Search**
- d. Topological Sort**

## **VII. Weighted Graphs**

- a. Dijkstra's**
- b. Prim's**
- c. Kruskal's**
- d. Network Flow**

## **VII. Divide and Conquer**

- a. Integer Multiplication**
- b. Tromino "Tiling"**
- c. Skyline Problem**
- d. Closest Pair of Points**

## **VIII. Dynamic Programming**

- a. Fibonacci**
- b. Combinations**
- c. Longest Common Subsequence**
- d. Number of Ways to Make Change**
- e. Fewest Number of Coins to Make Change**
- f. 0-1 Knapsack Problem**
- g. Floyd-Warshall's Algorithm and path reconstruction**
- h. Matrix Chain Multiplication**
- i. Edit Distance**

## **IX. Odds and Ends**

- a. Bellman-Ford (should have been taught with Dijkstra's)**
- b. Rolling Hash Function Calculation and String Matching**