

Midterm II Study Sheet

Emphasis is on Greedy Algorithms and Graphs: Formulation, Application, Performance with far less emphasis on writing psuedocode (meaning at most 1 problem requiring psuedocode) than on the first exam.

1. Greedy Algorithms
 - a. Activity Selection Problem
 - b. Huffman Code
 - c. Fractional Knapsack Problem
 - d. Dijkstra's Algorithm
 - e. Kruskal's Algorithm
 - f. Prim's Algorithm
2. Graph Concepts & Algorithms
 - a. Breadth First Search
 - b. Depth First Search
 - c. Topological Sort
 - d. Strongly Connected Components
 - e. DAG
3. Weighted Graphs
 - a. Single Source Shortest Paths
 - b. Bellman-Ford Algorithm
 - c. Topological Sorted DAG Algorithm
4. Performance and Proofs
 - a. Correctness of Greedy Algorithm (Huffman code example)
 - b. Correctness of Graph Algorithms (Chapt 22)
 - c. Performance of various Algorithms : $T(n)$ and $Size(n)$
 - d. Minimal Spanning Tree (Generic Algorithm Proof)

Points to consider:

1. What assumptions are built into the algorithms (if any) ?
2. What information does each algorithm require (generate) ?
3. Why do greedy algorithms perform better than dynamic programming solutions?
Under what circumstances do greedy algorithms fail?