Syllabus for Midterm Exam 2

CS/ECE 374-B Introduction to Algorithms and Models of Computation

University of Illinois Urbana-Champaign

Spring 2024

The second midterm exam will test material covered in lectures 9 through 15.

Specific skills that may be tested include (the following list may not be exhaustive):

1. Divide and Conquer Paradigm
   (a) Solving recurrences characterizing the running time of divide and conquer algorithms
   (b) Familiarity with specific Divide and Conquer Algorithms and the running times: Binary Search, Merge Sort, Quick Sort, Karatsuba's Algorithm, Linear Selection
   (c) Ability to design and analyze divide and conquer algorithms for new problems
2. Dynamic Programming Algorithms
   (a) Using the dynamic programming methodology to design algorithms for new problems
   (b) Ability to analyze the running time of dynamic programming algorithms
3. Graphs
   (a) Basic definitions of undirected and directed graphs, DAGs, paths, cycles.
   (b) Definitions of reachable nodes, connected components, and strongly connected components
   (c) Understand the structure of directed graphs in terms of the meta-graph of strongly connected components
   (d) Understand the structure of DAGs: sources, sinks, and topological sort
   (e) Solving dynamic programming problems using problems on DAGs
4. Graph Search
   (a) Understand properties of the basic search algorithm and its running time
   (b) Understand properties of DFS traversal on a directed and undirected graph
   (c) Understand properties of the DFS tree
   (d) Algorithms based on search for finding connected components in undirected graphs, checking whether a graph is a DAG, topological sort for DAGs, knowledge of a linear-time algorithm to create the meta-graph, finding a cycle in a graph, etc
   (e) Algorithms for DFAs/NFAs using graph algorithms
5. Paths in Graphs
   (a) Understand properties of the BFS trees
   (b) Understand properties of BFS traversal on directed and undirected graphs to find distances in unweighted graphs
   (c) Dijkstra's algorithm for finding single-source shortest paths in undirected and directed graphs with non-negative edge lengths

(d) Negative length edges and Bellman-Ford algorithm to check for negative length cycles or find shortest paths if there are none

(e) Floyd-Warshall algorithm

(f) Single-source shortest paths in DAGs — linear time algorithm for arbitrary edge lengths.

(g) Shortest path trees and their basic properties

(h) Dynamic programming for shortest path problems in graphs