Syllabus for Midterm Exam 3

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

University of Illinois Urbana-Champaign

Spring 2024

The third midterm exam will test material covered in lectures 16 through 24 excluding lecture 18 on minimum spanning trees.

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

1. Shortest Paths in Graphs
   (a) Dijkstra's algorithm for finding single-source shortest paths in undirected and directed graphs with non-negative edge lengths
   (b) Negative length edges and Bellman-Ford algorithm to check for negative length cycles or find shortest paths if there are none
   (c) Floyd-Warshall algorithm
   (d) Single-source shortest paths in DAGs – linear time algorithm for arbitrary edge lengths.
   (e) Shortest path trees and their basic properties
   (f) Dynamic programming for shortest path problems in graphs
2. Graph reductions and tricks
   (a) Modeling problems via graphs and solving them using graph structure, reachability, and shortest path algorithms
   (b) Adding sources, sinks, splitting edges, nodes
   (c) Creating layered graphs
3. Reductions
   (a) Knowing how to reduce one problem to another
   (b) Understanding what each reduction direction proves
4. Undecidability
   (a) Knowledge that halting is undecidable
   (b) Ability to prove that problems on program behavior are undecidable via reductions from HALT
5. NP, NP-Completeness and Polynomial-time Reductions
   (a) Definitions of NP, NP-Complete, NP-Hard
   (b) Knowledge of standard NP-Complete problems: SAT, 3SAT, CircuitSAT, Independent Set, Clique, Vertex Cover, Hamiltonian Cycle/Path in directed/undirected graphs, 3Color, Color
   (c) Ability to prove that a given problem is in NP
   (d) Ability to prove that a given problem is NP-Hard via a polynomial time reduction from an existing NP-Hard problem from the given list

(e) Understand the definition of a polynomial-time reduction and its implications

(f) Ability to prove the correctness of reductions

(g) Understand basic boolean logic and properties of SAT/CircuitSAT formulas to enable reductions