Exam 3 Review

csci2200, spring 2024, Bowdoin College

Topics covered: Graphs.

Resources: Lecture notes, all graph labs and assignments (Assignment 9, 10, 11). Any problem on the assignments and labs is fair game. When you review, rather than reading the solutions (not very effective), try closing the notes, solve the problem on your own without looking at the notes, and then compare to the notes.

The exam is not cumulative to the extent possible, but: in order to understand graph algorithms you need to know concepts learnt before graphs (e.g. asymptotic analysis, recurrences, etc) as well as understand techniques like greedy and dynamic programming.

1. Graph representation
    - adjacency list vs adjacency matrix
2. Basic graph concepts
    - undirectd graphs: connected components, connectivity, trees
    - directed graphs: reachability, strongly connected components
    - complete graphs
3. Graph traversal: BFS and DFS
    - undirected G: BFS(s)/ DFS(s) visit the connected component of s and run in linear time (in the size of the visited graph)
    - directed G: BFS(s)/ DFS(s) visit the vertices reachable from s and run in linear time (in the size of the visited graph)
    - extended to BFS/DFS the whole graph G
    - BFS applications include: connected components, paths, reachability, shortest paths (G unweighted, all edges have same weight), bipartiteness (G undirected)
    - DFS applications include: connected components, paths, reachability, directed cycles, topological sort (if G is a DAG)
4. Directed acyclic graphs (DAGs)
    - Topological order: Order of the vertices such that for any edge (x, y), vertex x comes before vertex y in topological order (all edges are “forward”)
    - Any DAG can be topologically ordered in $O(V + E)$ time by repeated sink removal, or via DFS

- Many problems have easier/faster solutions on DAGs. E.g. SSSP on DAGs can be computed in$O(V + E)$ time; longest paths on DAGs in$O(V + E)$ (on general graphs longest path is NPC).

5. Single-source shortest paths SSSP(s): Find shortest paths from a vertex s to all other vertices

   - Unweighted graphs: $O(V + E)$ time with BFS
   - DAGs: $O(V + E)$ time (one round of edge relaxation, in topological order)
   - General graphs with non-negative weights: Dijkstra's algorithm in $O(E \lg V)$ time (one round of edge relaxation with vertices in increasing order of distance from the source)
   - General digraphs without negative cycles: Bellman-Ford's algorithm in $O(V \cdot E)$ (run $O(|V| - 1)$ rounds of edge relaxation
   - Does the graph contain a negative cycle? run Bellman-Ford one more round