COMP 340 **Analysis of Algorithms**

Problem Set 05

Assigned: April 05, 2024 *Due: April 19, 2024*

1. 7.9

   Present correct and efficient algorithms to convert an undirected graph G between the following graph data structures. Give the time complexity of each algorithm, assuming $n vertices and m edges.

   (a) Convert from an adjacency matrix to adjacency lists.

   (b) Convert from an adjacency list representation to an incidence matrix. An incidence matrix M has a row for each vertex and a column for each edge, such that $M[i,j] = 1$ if vertex i is part of edge j, otherwise $M[i,j] = 0$.

   (c) Convert from an incidence matrix to adjacency lists.

2. 7-10

   Suppose an arithmetic expression is given as a tree. Each leaf is an integer and each internal node is one of the standard arithmetical operations ($+$, $-$, $*$, $/$). For example, the expression $2+3*4+(3*4)/5$ is represented by the tree in Figure 7.17(a) (p. 237). Give an $\mathcal{O}(n)$ algorithm for evaluating such an expression, where there are n nodes in the tree.

3. 7-17 a,b

   A *vertex cover* of a graph $G = (V, E)$ is a subset of vertices V' such that each edge in E is incident to at least one vertex of V'.

   (a) Give an efficient algorithm to find a minimum-size vertex cover if G is a tree.

   (b) Let $G = (V, E)$ be a tree such that the weight of each vertex is equal to the degree of that vertex. Give an efficient algorithm to find a minimum-weight vertex cover of G.

   (c) [BONUS] Let $G = (V, E)$ be a tree with arbitrary weights associated with the vertices. Give an efficient algorithm to find the minimum-weight vertex cover of G.

4. 7-20

   A vertex cover of a graph $G = (V, E)$ is a subset of vertices V' such that each edge in E is incident on at least one vertex of V'. An independent set of graph $G = (V, E)$ is a subset of vertices $V' \in V$ such that no edge in E contains both vertices from V'.

   An independent vertex cover is a subset of vertices that is both an independent set and a vertex cover of G. Give an efficient algorithm for testing whether G contains an independent vertex cover. What classical graph problem does this reduce to?

5. 7-22

   Consider a set of movies $M_1, M_2, \ldots, M_k$. There is a set of customers, each one of which indicates the two movies they would like to see this weekend. Movies are shown on Saturday evening and Sunday evening. Multiple movies may be screened at the same time.

   You must decide which movies should be televised on Saturday and which on Sunday, so that every customer gets to see the two movies they desire. Is there a schedule where each movie is shown at most once? Design an efficient algorithm to find such a schedule if one exists.

6. LeetCode 7-3 (Course Schedule)

   There are a total of `numCourses` courses you have to take, labeled from `0` to `numCourses - 1`. You are given an array prerequisites where `prerequisites[i] = [ai, bi]` indicates that you must take course `bi` first if you want to take course `ai`.

   For example, the pair `[0, 1]`, indicates that to take course `0` you have to first take course `1`.

   Return `true` if you can finish all courses. Otherwise, return `false`.

7. 8-1 a-c

   For the graphs in Problem 7-1:

   (a) Draw the spanning forest after every iteration of the main loop in Kruskal's algorithm.
   (b) Draw the spanning forest after every iteration of the main loop in Prim's algorithm.
   (c) Find the shortest-path spanning tree rooted in A.

8. 8-2

   Is the path between two vertices in a minimum spanning tree necessarily a shortest path between the two vertices in the full graph? Give a proof or a counterexample.

9. 8-15

   The *single-destination shortest-path* problem for a directed graph seeks the shortest path from every vertex to a specified vertex v. Give an efficient algorithm to solve the single-destination shortest-path problem.

10. 8-19

    Give an efficient algorithm to find the shortest path from x to y in an undirected weighted graph $G = (V, E)$ with positive edge weights, subject to the constraint that this path must pass through a particular vertex z.