Algorithms -1
Tutorial 6
October 26, 2018

For all the problems, first solve assuming that the graph is represented using an adjacency list. Then think what the time complexity of your algorithm will be if the graph is represented by an adjacency matrix.

1. Prove or disprove: BFS and DFS algorithms on a undirected, connected graph G = (V,E) produce the same tree if and only if G is a tree.
2. There are N cities in Magicland. Some of the cities are connected by some roads. A road connects two cities and is bi-directional i.e., we can go either way through a road. There is a path from a city i to every other city j and there is no cycle among the cities. You need to find a pair of cities (i, j) such that the length of the path between i and j is maximum among all such pairs. The length of a path is the number of edges on the path. Your algorithm should run in O(V + E) time.
3. Consider a directed unweighted graph *G*. Suppose that G has *k* strongly connected components $S_1, S_2, \ldots, S_k$. We form another directed graph G_1 with *k* nodes, numbered from 1 to *k*, with an edge from node *i* to node *j* if and only if there is an edge from some node in S_i to some node in S_j in G. What is the maximum possible number of cycles in G_1? Justify your answer clearly.
4. You are given a DAG (Directed Acyclic Graph) G(V,E). You need to find the longest path in G. The length of a path is the number of edges on the path. Your algorithm should run in O(V + E) time.
5. Give an O(V + E) time algorithm that takes as input a directed acyclic graph G = (V,E) and two nodes s and t, and returns the number of paths from s to t in G.
6. There are N variables $x_1, x_2, \ldots, x_N$ and M relations of the form $x_i < x_j$ where $i \neq j$. A subset S of relations is called inconsistent if there does not exist any assignment of variables that satisfies all the relations in S. e.g. $\{x_1 < x_2, x_2 < x_1\}$ is inconsistent. You need to find if there's an inconsistent subset of M. Your algorithm should run in O(V + E) time.
7. Let $G = (V, E)$ be an undirected graph represented as an adjacency list. A subset of edges F is called a ***feedback edge set*** if F contains at least one edge of every cycle of G. The size of a feedback edge set is the number of edges in it. A minimum size feedback edge set is a feedback edge set with minimum size among all possible feedback edge sets of G. Design an $O(|V|+|E|)$ time algorithm to find a minimum size feedback edge set of G.
8. The eccentricity of a vertex v in a graph is the maximum distance from v to any other vertex in the graph. The center of a graph is the set of all vertices with minimum eccentricity. Your task is to find the center of a tree in linear time.
9. You are given a graph G = (V, E) with positive edge weights, and a minimum spanning tree $T = (V, E_1)$ with respect to these weights. Now suppose the weight of a particular edge e in E is modifed from w(e) to a new value w'(e). You wish to quickly update the minimum spanning tree T to reflect this change, without recomputing the entire tree from scratch. There are four cases. In each case give a linear-time algorithm for updating the tree, if needed.
    (a) e not in E_1 and w'(e) > w(e).
    (b) e not in E_1 and w'(e) < w(e).
    (c) e in E_1 and w'(e) < w(e).
    (d) e in E_1 and w'(e) > w(e).