CS 344 Problem Set 3: Graphs - Structure and Connectivity

Due : (August 6, 2023 11:55 pm)

The objective of this assignment is to get you to work on developing graph algorithms and proving properties about graph structures.
You must attempt all problems, even if you are unable to completely solve them.
You may discuss these problems with your classmates - and you are strongly encouraged to discuss them with me.

1. (**Bipartite graphs**) An undirected graph $G = (V, E)$ is called a bi-partite graph if its vertex set V can be split into two *disjoint* groups L, R such that all edges have exactly 1 end point in each of these two sets (in other words $E \subseteq L \times R$). Prove that the following statements are equivalent :

   (a) G is a bi-partite graph.

   (b) The vertices of G can be colored using only two colors such that every edge is between vertices of opposite colors.

   (c) G has no odd-length cycles.

   In order to prove the equivalence of these statements show that $(a) \Leftrightarrow (b)$ and $(a) \Leftrightarrow (c)$. Then, provide a linear time algorithm to determine whether a given graph is bi-partite.

2. (**Binary tree exploration**) You are given as input a rooted binary tree $T = (V, E)$ with root $r \in V$ (a binary tree is a tree where every vertex has no more than 2 children). Vertex u is said to be an *ancestor* of vertex v if the path from root r to vertex v passes through u. Provide linear time algorithms for the following tasks :

   (a) (Ancestor queries) Given any pair $(x, y) \in V \times V$ - answer YES if x is an ancestor of y in T and NO otherwise. (Run an algorithm to pre-process the graph in linear time and then set up information such that ancestors queries can then be answered in $O(1)$ time)

   (b) Given as input a rooted binary tree $T = (V, E)$ with root $v_1 \in V$ where each vertex $v \in V$ is given a value $s(v)$. Define $a(v)$ as follows :

   $$a(v) = \text{the maximum of all s-values among the descendants of } v.$$

   Give a linear time algorithm that calculates the array A where $A[v_i] = a(v_i)$ (you may assume that the vertices are numbered $v_1, v_2, .., v_n$ and v_1 is the root).

3. (**Dikjstra's "fix"**) Consider the following "fix" to Dijkstra's algorithm for graphs with negative edges : On input graph $G = (V, E)$ with weights $w : E \to \mathbb{Z}$ (assume all weights are integers - some possibly negative) first find the weight of the largest edge (say max). Next, add max to the weights of all edges so that they have non-negative weights. Apply Dijkstra's algorithm on these weights to get the shortest paths from $s \in V$ to all other vertices of G.

   (a) What is the time complexity of this algorithm? Assume that a binary heap is used for Dijkstra's algorithm.

   (b) Prove/Disprove that this fix results in actually finding the shortest paths from vertex s to all other vertices. It will be helpful to try this "fix" out on a couple of example weighted graphs.

4. (**Tank capacity**) You are given a set of V cities. Some of these cities have two-way highways connecting them. Each highway $e \in E$ has length $l_e \in \mathbb{Z}^+$ in miles. Each city has a gas station, but none of the highways do. Your car has a fuel tank capacity to cover L miles. So you can only go from city u to city v if $l_{uv} \leq L$.

(a) Provide a linear time algorithm that determines whether or not your car can travel from city s to city t.

(b) Provide an efficient algorithm that computes the minimum tank capacity needed to travel from city s to city t.

5. (**Extra Credit**) Given an undirected graph (with all edge weights 1) and two vertices (s, t), provide a linear time algorithm to compute the number of distinct shortest paths from s to t.