(CS1.301) Algorithm Analysis & Design (Monsoon 2024)

Practice Problem Set 1

Instructions:

- The following problem set is not graded and is for practice.
- Some of the problems will be covered in the tutorial.

Question 1

We have a connected graph $G = (V, E)$ and $u \in V$ be a specific vertex. Suppose we compute a depth-first search (DFS) rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search (BFS) tree rooted at u, and obtain the same tree T. We want to prove that $G = T$. That is, if T is both a DFS tree and BFS tree rooted at u, then G cannot contain any more edges than those in T.

Question 2

There is a natural intuition that two nodes that are far apart in a communication network separated by many hops have a more tenuous connection than two nodes that are close together. There are a number of algorithmic results that are based to some extent on different ways of making this notion precise. Here's one that involves the susceptibility of paths to the deletion of nodes.

Suppose that an n-node undirected graph $G = (V, E)$ contains two nodes s and t such that the distance between s and t is strictly greater than $\frac{n}{2}$. Show that there must exist some node v, not equal to either s or t, such that deleting v from G destroys all s to t paths. (In other words, the graph obtained from G by deleting v contains no path from s to t.) Give an algorithm with running time $O(m + n)$ to find such a node v.

Question 3

A vertex u in a connected (undirected) graph $G = (V, E)$ is called an articulation point if the removal of u from the vertex set, and the removal of edges incident on the vertex u disconnects the graph. Give an algorithm to find articulation point(s) in a given graph.

Question 4

A directed graph $G = (V, E)$ is strongly connected if and only if every pair of vertices is strongly connected. Equivalently, a strong component of G is a maximal strongly connected subgraph of G. A directed graph G is strongly connected if and only if G has exactly one strong component; at the other extreme, G is a DAG if and only if every strong component of G consists of a single vertex. Give an algorithm that computes all strong components of a graph in time at most $O(V \cdot E)$.
Bonus: Can this be improved to $O(V + E)$?

Question 5

Recall that a directed graph G is strongly connected if, for any two vertices u and v, there is a path in G from u to v and a path in G from v to u.

Describe an algorithm to determine, given an undirected graph G as input, whether it is possible to direct each edge of G so that the resulting directed graph is strongly connected.

Question 6

A graph (V, E) is bipartite if the vertices V can be partitioned into two subsets L and R, such that every edge has one vertex in L and the other in R.

(a) Prove that every tree is a bipartite graph.

(b) Prove that a graph G is bipartite if and only if every cycle in G has an even number of edges.

(c) Describe and analyze an efficient algorithm that determines whether a given undirected graph is bipartite.