(CS1.301) Advanced Algorithm Design (Monsoon 2022) Due: 24.08.2022

Problem Set 1

Instructions:

- Discussions amongst the students are not discouraged, but all writeups must be done individually and must include names of all collaborators.
- Referring sources other than the lecture notes is discouraged as solutions to some of the problems can be found easily via a web search. But if you do use an outside source (eg., text books, other lecture notes, any material available online), do mention the same in your writeup. This will not affect your grades. However dishonesty of any sort when caught shall be heavily penalized.
- Be clear in your arguments. Vague arguments shall not be given full credit.
- Total marks are 30 but they will be normalized.

Question 1 Consider the following functions.

$$T_1(n) = a \cdot T_1\left(\frac{n}{b}\right) + b \cdot n$$
$$T_2(n) = b \cdot T_2\left(\frac{n}{a}\right) + a \cdot n$$

If $a \geqslant b$ and $T_1(1) = T_2(1) = 1$, how do these functions compare as n grows large. [2 marks]

Question 2 In the class we saw an algorithm to detect cycles in directed graphs. Give an algorithm to detect cycles in undirected graphs. [4 marks]

Question 3 A binary tree is a rooted tree in which each node has at most two children. Show by any means possible that in any binary tree the number of nodes with two children is exactly one less than the number of leaves. [4 marks]

Question 4 We have a connected graph $G = (V, E)$, and a specific vertex $u \in V$. Suppose we compute a depth-first search rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search tree rooted at u, and obtain the same tree T. Prove that $G = T$. That is, if T is both a depth-first tree and breadth-first tree rooted at u, then G cannot contain any more edges than those in T. [6 marks]

Question 5 Prove or disprove the following claim. Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected. [6 marks]

Question 6 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 $n/2$. 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. [8 marks]