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 for this problem set are 20.

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

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. [5 marks]

Question 3

A binary tree is a rooted tree in which each node has at most two children. In a binary tree, a leaf is defined as a node with no children, and an internal node is a node with at least one child.

(a) Show that in any binary tree the number of nodes with two children is exactly one less than the number of leaves. [4 marks]

(b) Suppose you have a full binary tree, where every internal node has exactly two children. Derive a formula that relates the number of leaves (L) to the total number of nodes (N) in such a binary tree. [2 marks]

Question 4

Let us define a generic graph traversal algorithm called Whatever-First Search(WFS) algorithm. The generic traversal algorithm stores a set of candidate edges in some data structure called a "bag". The only important properties of a "bag" is that we can put stuff into it and then later take stuff back out. For example, a stack is a particular type of bag, but not the only one.

```
WhateverFirstSearch(s):
  put s into the bag
  while the bag is not empty
      take v from the bag
      if v is unmarked
          mark v
          for each edge vw
              put w into the bag
```

Figure 1: Pseudocode for WFS Algorithm

(a) Prove that WhateverFirstSearch(s) marks every vertex reachable from s and only those vertices(i.e. no other vertices apart from the ones which are reachable from s). [2 marks]

(b) Now, instead of maintaining marked and unmarked vertices as in the case of WFS Algorithm, we define an algorithm which maintains a color for each vertex, which is either white, gray, or black. In the following algorithm, imagine a fixed underlying graph G.

```
ThreeColorSearch(s):
    color all nodes white
    color s gray
    while at least one vertex is gray
        ThreeColorStep( )
```

```
ThreeColorStep( ):
    v ← any gray vertex
    if v has no white neighbors
        color v black
    else
        w ← any white neighbor of v
        parent(w) ← v
        color w gray
```

Figure 2: Pseudocode for ThreeColorSearch Algorithm

(i) Prove that ThreeColorSearch(s) maintains the following invariant at all times: No black vertex is a neighbor of a white vertex. [2 marks]

(ii) Prove that after ThreeColorSearch(s) terminates, all vertices reachable from s are black, all vertices not reachable from s are white, and that the parent edges $v \to \text{parent}(v)$ define a rooted spanning tree of the component containing s. [3 marks]