ESO207A: **Data Structures and Algorithms**	
Homework *4b*	Due Date: 13 November 2017

Instructions.

1. For each problem, write your name, Roll No., the problem number, the date and the names of any students with whom you collaborated.
2. For questions in which algorithms are asked for, give (a) a clear description of the algorithm in English and/or pseudo-code, (b) A proof/argument of the correctness of the algorithm, and, (c) an analysis of the running time of the algorithm.

Full marks will be given only to correct solutions which are described clearly. Convoluted and unclear descriptions will receive *low marks*.

Problem 1. Let $G = (V, E)$ with an undirected graph with non-negative edge weights $w_e \geq 0$. Suppose T is a minimum spanning tree of G and G_π is a single source shortest path tree with root s (say, it is the result of running Dijkstra's algorithm from source s).

(a) Can T and G_π be the same? Can they be different? Give examples in each case.

Suppose each edge weight is incremented by 1: $w'_e = w_e + 1$.

(b) Does the minimum spanning tree change? Given an example where it changes or prove that it does not change.

(c) Does the shortest path tree change? Given an example where it changes or prove that it does not change.

Problem 2. Let $G = (V, E)$ be an undirected graph with all *distinct* edge weights. Prove that it has a unique minimum spanning tree.

Problem 3. Let $G = (V, E)$ be an undirected graph with non-negative (real valued) edge weights. Many of these edge-weights are integral. You have to design an efficient algorithm to find the *minimum integral spanning tree.* An integral spanning tree is a set T of edges that forms a spanning tree and the edge-weights of all edges in T are integral. A minimum integral spanning tree is an integral spanning tree whose cost (sum of edge-weights in T) is no larger than the cost of every integral spanning tree. The algorithm should find the minimum integral spanning tree if one exists and should return *NO* if there is no integral spanning tree.

Problem 4. [CLRS 16-1] Consider the problem of making change for n Rupees (n positive integral) using the fewest number of coins with denominations 10, 5, 2 and 1.

(a) Describe a greedy algorithm for this specific problem. Prove that your algorithm yields an optimal solution.

(b) Give a collection of coin denominations where the greedy algorithm does not yield an optimal solution. The collection must include the Re 1 coin so that there is a solution for every value of n.