CSC 3102 Fall 2019
Homework 4
due date 12/3/19, 80 points

1. (20 points) We define a following *temporal graph.* The vertex set of the graph is set of airports given by their 3 letter airport code. The edge set, representing flights, consists of edge-object which is is a 5-tuple $< FLT\_NO, ORG, DST, DEPT\_TIME, ARR\_TIME >$. These indicate flight number, origin airport, destination airport, scheduled departure time and arrival time. Note that this temporal graph could have multiple parallel edges between same airport pairs - since each edge could have be different flight between the same airport pair. Now we given such a graph and then we are given two airports START and END and we are given a start time t. Our task is to find fastest itnerary to reach airport END as early as possible starting from airport START after time t. This itinerary is likely to involve multiple connections. We must leave a gap of one hour between connecting flights at any airport.

   Design an efficient algorithm to compute such an itinerary. Give pseudocode and analyse its complexity.

   Hint: Modify Dijkstra's algorithm for shortest path problem in graphs.

2. (20 points) Vertex Cover Problem on Trees: Given a tree $T(V, E)$ with positive weight on its vertices $w : V \to R^+$. Thus, each vertex v has a positive weight $w(v)$ associated with it. Now, we need to find a minimum weight subset $C \subseteq V$ (i.e., sum of weights of vertices in this subset needs to be minimum) subject to constraint that for every edge in the tree, at least one of its endpoints should appear in this subset C. If we imagine a tree as being rooted than this means that for any node, either it should be present in the subset or its parent (or both).

   Design an efficient algorithm to compute such a vertex cover. Give pseudocode and analyse its complexity.

   Hint: Modify the dynamic programming algorithm for Independent set problem accordingly.

3. (40 points) For ONE of the above two problems (of your choice) - write a program in C/C++ or Java. For this, you will create a small (with 10 to 20 items) example of the problem. Draw it on paper. Use this as a sample input and run your code. Verify if your answer matches with what you have on the paper. You will not submit the code on classes server for this. You will take a printout of your code and printout of your output and attach it along with your work on the paper.