Data Structure HW#6

Part I Paper Work

1 (Exercise 5 of Chapter 6.1) Obtain the adjacency-matrix, adjacency-list, and adjacency-multilist representations of the graph of Figure 6.16.

2 (Exercise 6 of Chapter 6.4) Use *ShortestPath* (Program 6.8) to obtain, in nondecreasing order, the lengths of the shortest paths from vertex 0 to all remaining vertices in the digraph of Figure 6.34.

3 (Exercise 8 of Chapter 6.4) Modify *ShortestPath* (Program 6.8) so that it obtains the shortest paths, in addition to the lengths of these paths. What is the computing time of your modified function?

4 (Exercise 2 of Chapter 6.5) (a) For the AOE network of Figure 6.44, obtain the early, *e*(), and late, *l*() start times for each activity. Use the forward-backward approach.
   (a) What is the earliest time the project can finish?
   (b) Which activities are critical?
   (c) Is there any single activity whose speed-up would result in a reduction of the project length?

Part II Programming Exercise

1 Implement the Graph ADT in C++. The Graph ADT contains at least the following methods:
   (a) InsertEdge(int u, int v)
   (b) InsertVertex(int v)
   (c) MinSpanTree()
   (d) ShortestPath(int v)

   Output:

   1) Take the digraph of Figure 1 as the input and output the minimal cost spanning tree by printing all the edges and also the total cost (total cost is the summation of the weight of all edges in the tree)

0, 2

2, 3

…

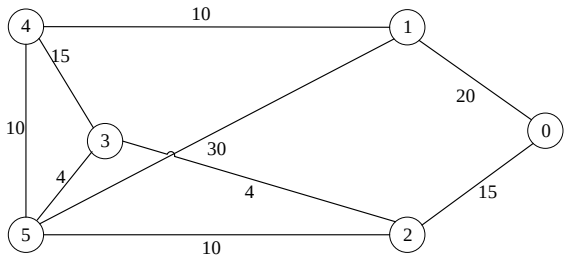

Figure 1

2) Take the digraph of Figure 2 as the input and find the shortest path from a source (say, 0) to all destinations, like

0, 2, 15

0, 1, 20

…

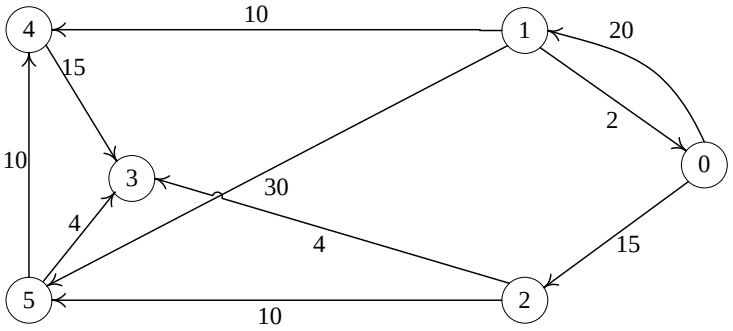

Figure 2

Note 1: You can reuse one-side used papers but must in A4 size. Please hand in your assignments to the TAs (R721, Applied Science &Technology Building) by the deadline.

Note 2: Please compress your code as well as the snapshot your execution results into a zip file and upload it to **iLearning**.