Assignment - 3

Total Points: 60

Instructions

Submission

You can either write your answer using Word/LaTex or take a picture/scan of your hand-written (neat and tidy) solution, then put your solution in one PDF file. Submit your PDF online using Blackboard. All submission must be posted before the deadline.

Problem 1 (Max Points:5)

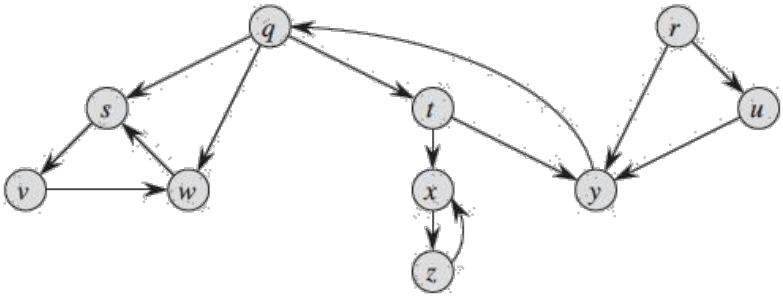
Given an adjacency-list representation of a directed graph, how long does it take to compute the out-degree of every vertex? How long does it take to compute the in-degrees? Justify your answer with proper reasoning.

Problem 2 (Max Points: 10)

The transpose of a directed graph G = (V, E) is the graph $G^T = (V, E^T)$, where $E^T = \{(v,u) \in V \times V : (u,v) \in E\}$ Thus, G^T is G with all its edges reversed. Describe efficient algorithms for computing G^T from G, for both the adjacency-list and adjacency-matrix representations of G. Analyze the running times of your algorithms.

Problem 3 (Max Points:10)

Show how depth-first search works on the graph below. Assume that the vertices are explored in alphabetical order (at any point of time if it has to be chosen between q and r, then q will be explored before r), and that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and also show the classification of each edge (Tree/Forward/Back/Cross).



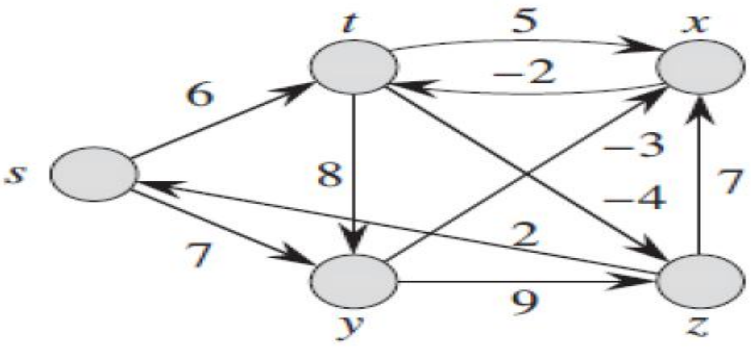
Problem 4 (Max Points:10)

Suppose that we represent the graph G = (V, E) as an adjacency matrix. Give a simple implementation of Prim's algorithm for this case that runs in $O(V^2)$ time.

Problem 5 (Max Points:15)

Run the Bellman-Ford algorithm on the directed graph given below, using vertex z as the source. In each pass, relax edges in the order (t,x),(t,y),(t,z),(x,t),(y,x), (y,z),(z,x),(z,s),(s,t),(s,y) and show the d and π values after each pass. (For reference: if (u,v) is an edge which is relaxed in the pass , then v. = u)

Now, change the weight of edge (z,x) to 4 and run the algorithm again, using s as the source.



Problem 6 (Max Points:10)

Dijkstra's Algorithm

```
DIJKSTRA(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = G.V
4  while Q ≠ ∅
5      u = EXTRACT-MIN(Q)
6      S = S ∪ {u}
7      for each vertex v ∈ G.Adj[u]
8          RELAX(u, v, w)
```

Run Dijkstra's algorithm on the directed graph given below, using vertex s as the source. Show the d and π values and the vertices in set S after each iteration of the while loop in the algorithm given above.