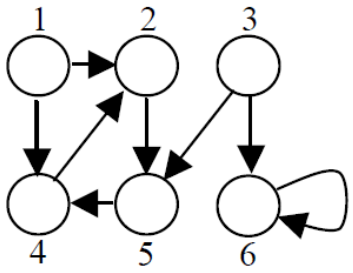
Helwan University
Faculty of Computers and Information
Computer Science Department

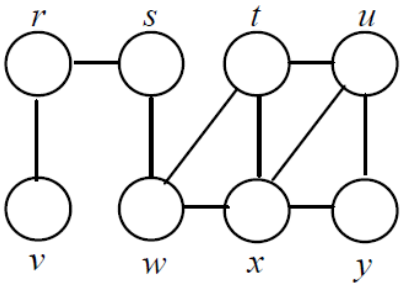
Graph HW#5

Q1- *Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap.

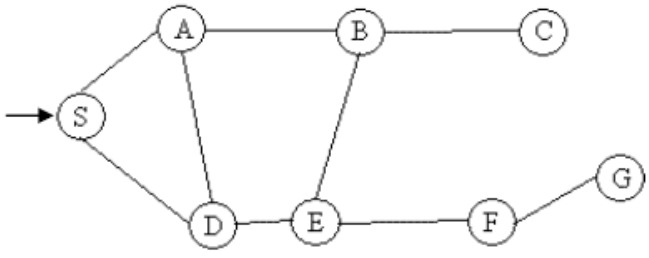
Q2- Show the d and π values that result from running breadth-first search on the following directed graph, using vertex 3 as the source.



Q3- *Show the d and π values that result from running breadth-first search on the following undirected graph, using vertex u as the source.



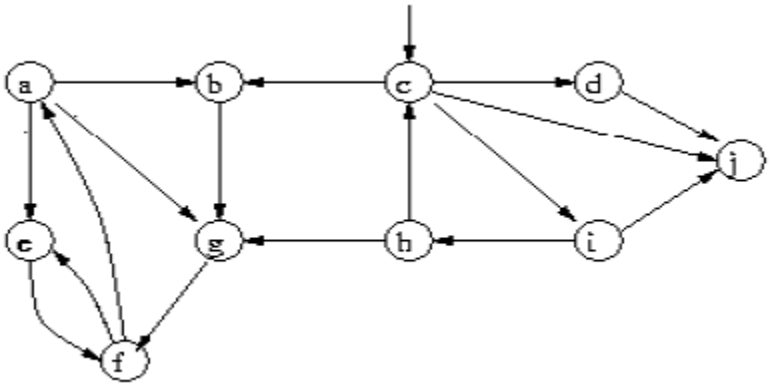
Q4- Show the result of running breadth-first search on the undirected graph:



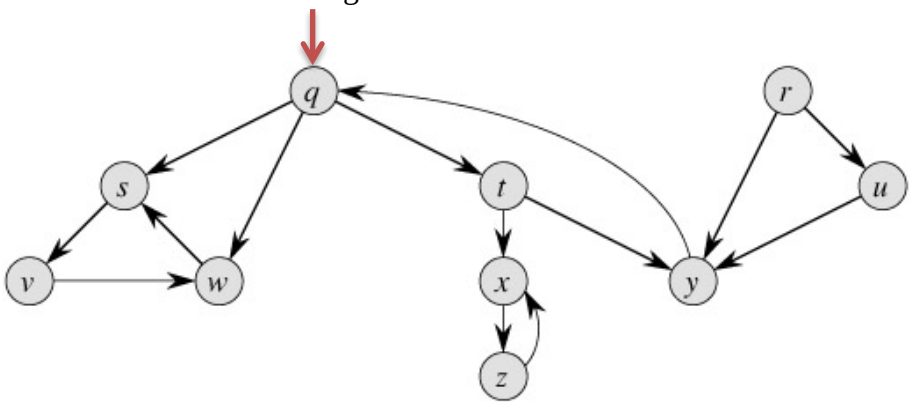
Q5- *Rewrite the procedure DFS, using a stack to eliminate recursion.

Q6- Modify the pseudo-code for depth-first search so that it prints out every edge in the directed graph *G*, together with its type. Show what modifications, if any, must be made if *G* is undirected.

Q7 – Show the result of running depth-first search on the directed graph:



Q8- *Show how depth-first search works on the following graph. Assume that the **for** loop of lines 5-7 of the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically. Show the discovery and finishing times for each vertex, and show the classification of each edge.



Q9- What is the running time of BFS if we represent its input graph by an adjacency matrix and modify the algorithm to handle this form of input?

Q10-* Let $G = (V, E)$ be an undirected graph. Give an $O(V + E)$-time algorithm to compute a path in *G* that traverses each edge in *E* exactly once in each direction. Describe how you can find your way out of a maze.