Assignment 5:

Due Date: Friday March 18th, 11:59 p.m.

Objectives:

- Graph algorithms

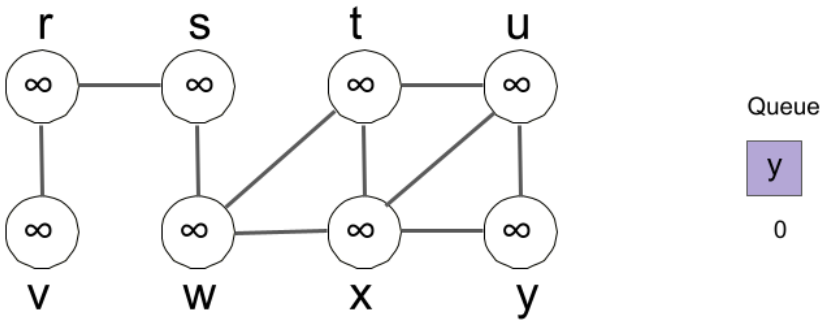
Submission Instruction

- **Start early**
- You are allowed to work in groups of at most two students. It is okay if you want to work on your own.
- Write your answers in a file named **A5.pdf**
- Make sure to have one submission per group
- Make sure that you are typing or your writing is neat and you follow submission instruction
- **Have your name(s) at the top of the pdf file and when you are submitting in MyLearningSpace**

Problems

[10 points] Q1.

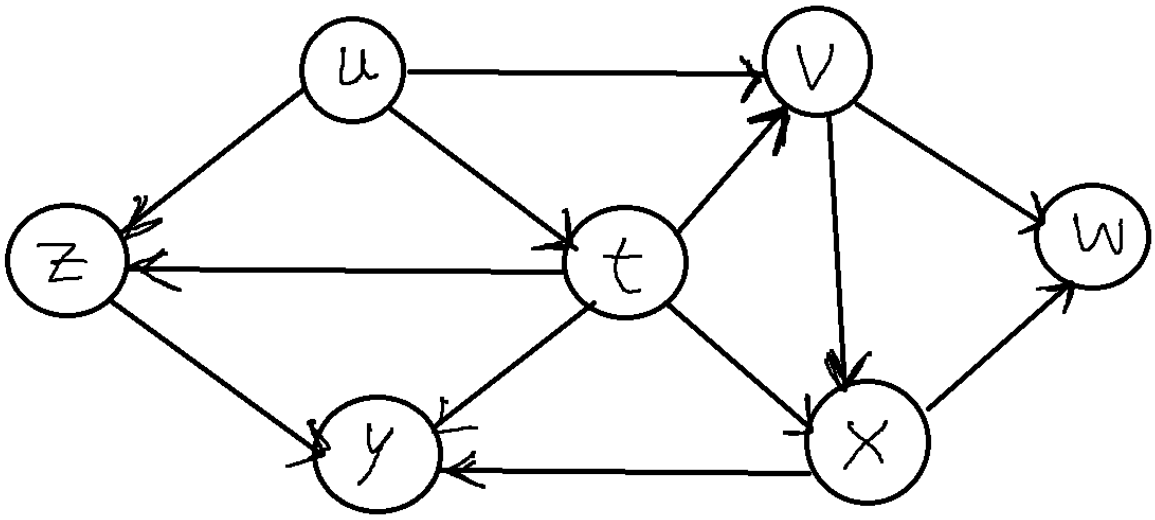
Run the BFS algorithm on the following graph and show the **pred** and **distance** arrays as the algorithm proceeds. Start from node **y**. Show the values on queue at each step of the algorithm.



[bonus–10 points] Q2.
For the same graph in question one, run DFS algorithm and show the **pred** and **discover** and **finish** arrays as the algorithm proceeds.

[10 points] Q3. You are given an undirected graph G=(V, E). Determine whether G is a tree or not? The runtime of your algorithm should be **O(V).** Write the pseudocode of your algorithm and analyze its time complexity.

[5 points] Q4. How many topological ordering does the following graph have? Write down all possible topological ordering.



[10 points] Q5. Prove that a directed acyclic graph must have at least one node u which has no incoming edge, i.e., indegree (u) = 0

Hint: proof by contradiction

[15 points] Q6. Given graph G = (V, E), the graph G'= (V', E') is defined such that (u,v) is in E' if and only if G contains a path with at most two edges between u and v.

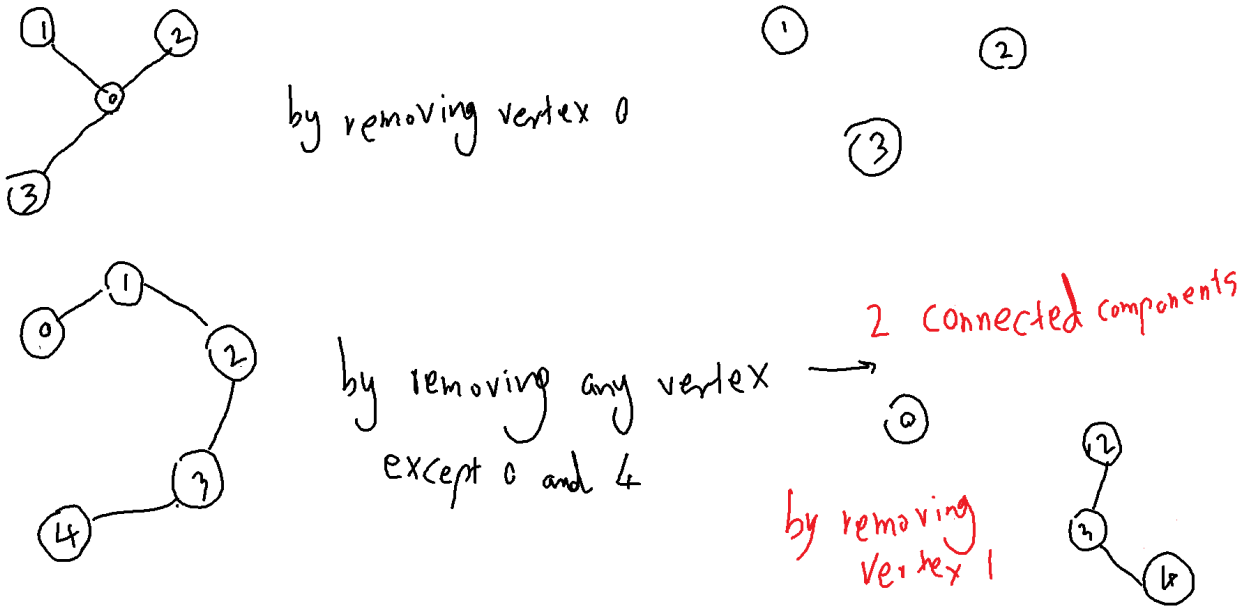
Describe an efficient algorithm for computing G' from G. Once develop the algorithm assuming G has an adjacency-matrix representation and once develop the algorithm assuming adjacency-list representation for the graph. Analyze the running times of your algorithms.

[10 points] Q7. Let G = (V, E) be an undirected graph. Let v be a vertex in G. Give an O(V+E) time algorithm that finds the shortest cycle in G which contains the vertex v.

[bonus–15 points]Q8.
You are given a tree with |V| vertices and |E| edges. Remember for a tree we have:
|E| = |V|-1

By removing one vertex and all the edges that go from that vertex to some other vertices, the tree will become disconnected and you end up with two or more connected components. Design an algorithm to return the largest number of components you can get by removing a single vertex of the tree. Develop the pseudocode and analyze the time complexity of your algorithm.

Example: The first one is a tree with four vertices and the second one is a tree with five vertices.



Grading

[Total points: 65]

[5 points] Following submission instruction
[10 points] Q1
[bonus-10 points] Q2
[10 points] Q3
[5 points] Q4
[10 points] Q5
[15 points] Q6
[10 points] Q7
[bonus-15 points] Q8