GRAPHS

Q1. G=(V,E) is a directed graph with 'n' vertices numbered 1, 2, … n. For every i<j, there is an edge (i, j) in G.

How many edges are there in G? ____n (n-1)/2_____

How many cycles are there in G? _____none_____

Q2. Let G=(V,E) be an undirected graph with *n* vertices. If G is connected and has *n-1* edges then G does not contain a cycle.

Is the above statement TRUE or FALSE? ____TRUE_____

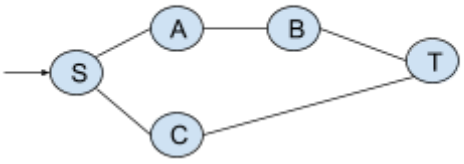
We know every n-node tree has exactly n-1 edges. The undirected graph G is connected, so it cannot contain a cycle.

Q3. Let $G=(V,E)$ be an undirected graph with n vertices. If there is *some* path p from vertex s to vertex t in G, then the tree returned by Breadth-First Search on G will contain p.

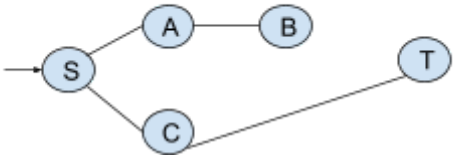
Is the above statement TRUE or FALSE? __FALSE_____

Brief explanation of your answer.

Graph G contains path p: S,A,B,T



Tree returned by BFS from start node S does not contain p



Q4. Given a **directed** graph **G(V,E)** with n vertices and m edges. Assume G is composed of a single component. Let **S** be the set of vertices that are reachable from vertex u in graph **G**. Graph $\mathbf{G^R(V,E)}$ is obtained by reversing the direction of all edges in graph **G**. Let $\mathbf{S^R}$ be the set of vertices that are reachable from vertex u in graph $\mathbf{G^R}$. If set S is same as set $\mathbf{S^R}$, what property can you conclude about the structure of **G**?

Property: Set S is a strongly connected component containing 'u'.

Give brief explanation of your answer:
A node has a path from ‘u’ in G^R iff it has a path to ‘u’ in G. If ‘u’ and ‘v’ are mutually reachable and ‘u’ and ‘w’ are mutually reachable, then ‘v’ and ‘w’ are mutually reachable.

Q5. Given a directed graph G(V, E), your friend ran Depth First Search (DFS) on it and found that it contains exactly one back-edge. Your friend now removes that back-edge from the graph G and claims that the resulting graph has a special property.

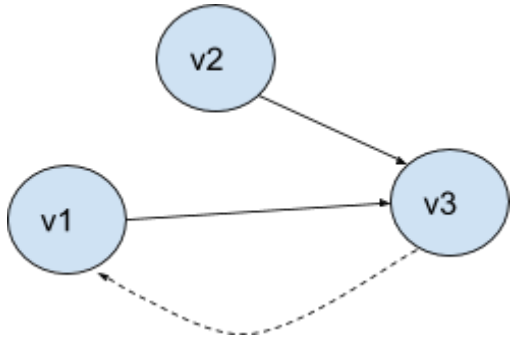
If AGREE, then explain what that special property is. If DISAGREE then mention why.
AGREE / DISAGREE ____AGREE_____

The resulting graph is a DAG.

Q6. Given a directed acyclic graph (DAG) G with at least n−1 edges. A topological sort of G gives the following ordering of vertices $v_1, v_2 \ldots, v_n$. If we add the edge (v_n, v_1) to G then the resulting graph is guaranteed to be strongly connected.

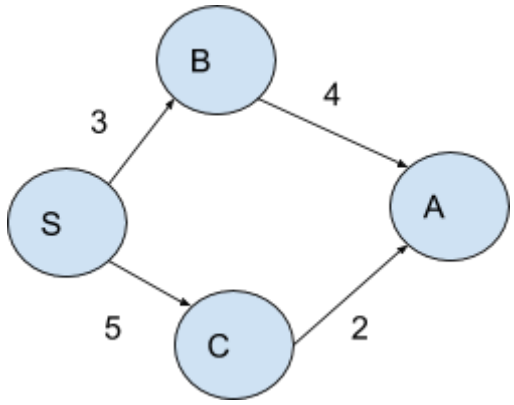
Is the above statement TRUE or FALSE? ____FALSE_____
Brief explanation of your answer.

Counterexample:



Q7. Given a graph G with positive weights on the edges. S and A are two vertices in G. If all the edge weights are **distinct**, then there is a **unique** shortest path from S to A.

Is the above statement TRUE or FALSE? ___FALSE_____
Brief explanation of your answer.
Counterexample:



Q8. Given an **undirected** connected graph `G(V,E)`. Is it possible that the depth of any DFS tree rooted at a vertex 'u' of `G` may be greater than the depth of any BFS tree rooted at 'u'. TRUE or FALSE ___TRUE_____ If TRUE, illustrate by giving an example of a graph `G` and the corresponding BFS and DFS trees. If FALSE, prove it.

This is an easy question. Think of an example.

Q9. Let's fast forward to your big graduation day! All of you have invited guests, who are going to be seated in a big hall. Suppose that guest Gx and Gy know each other and Gx and Gz know each other, then Gx can introduce Gy to Gz (so now Gy and Gz know each other indirectly). LUMS wants your guests to have a great time and has created some seating areas. They want people who know each other (**directly or indirectly**) to be seated together in one area so that they are comfortable talking to each other. You have to find the **minimum** number of seating areas for this purpose.

You have studied a number of algorithms in class. Can you use any of those algorithms to **efficiently** solve this problem? **Note:** You will **not** get marks if you try to design a new algorithm from scratch.

Finding the number of connected components in a graph.

_____

Briefly justify your answer.

The number of connected components in a graph can be found by DFS or BFS.

What is the time complexity of your algorithm in Big-Oh notation ___ O(m+n)_____

Q10. Given a **directed**, unweighted graph with no cycles. We want to find out if there is a path that visits **each** vertex **exactly once**. If there is a path, your algorithm should find it, else say that no such path is possible. Give a **linear time** algorithm for this problem.

Write your algorithm in clear descriptive steps as in Q1 above (do NOT write C code).

This is your next homework, which is due on Monday, Oct. 21 @ 11:00 AM in class.

Running time of your algorithm in Big-Oh notation _____
Briefly justify your answer.