CS 3510: Design and Analysis of Algorithms	Georgia Tech

Exam 2

Instructor: Shahrokh Shahi — *Summer 2022*

GT Username:	**Full Name:**

Instructions:

1. Write your **name** and **GT username** on each page very clearly. Then, complete the exam.
2. This exam is closed-book, and collaboration is NOT permitted.
3. You are allowed to use **one sheet of notes**, i.e., both sides of a letter-sized paper, during the exam.
4. No calculator is required.
5. You have **80 minutes** to complete this exam.
6. It is recommended to read all the questions before starting. Please read the questions carefully. Misunderstanding the question is not a valid excuse for losing points.
7. If you find it necessary, make reasonable assumptions but make sure to state them clearly.
8. You can use the back of each sheet as scratch paper.
9. Write your solution in the space provided. In case you need more space, you can use back of the same sheet, and make a notation on the front of the sheet.
10. The exam has 50+2 points in total.

Good luck!

Number	Problem	Points	Grade
1	Short Answers	20+2	
2	Graph Traversal Applications	10	
3	Minimum Spanning Tree	12	
4	Minimum Spanning Tree	8	

GT Username:	Full Name:

1 Short questions [20+2 pts]

(a) (8 pts) Complete the following table by writing the data structure and running time of each of these algorithms discussed in class.

Algorithm	Data Structure	Running Time
Breadth-first search (BFS)	Queue	$O(\lvert V\rvert + \lvert E\rvert)$ or $O(n+m)$
Depth-first search (DFS)	Stack	$O(\lvert V\rvert + \lvert E\rvert)$ or $O(n+m)$
Kruskal	Disjoint Set (Union-Find)	$O(\lvert E\rvert \log \lvert V\rvert)$ or $O(m \log n)$
Prim	Priority Queue (Min-Heap)	$O(\lvert E\rvert \log \lvert V\rvert)$ or $O(m \log n)$

(b) (8 pts) For each of the following statements, decide whether it is **T**rue or **F**alse. If it is true, provide a short explanation and if it is false, give a counterexample.

1. In the interval-scheduling (activity-selection) problem discussed in class, "earliest start time", i.e., considering jobs in ascending order of starting times, is the greedy choice that gives the optimum greedy solution. False; consider `intervals = [[1, 10], [2, 3], [4, 5], [7, 8]]`. The optimum solution is 3 but the given greedy choice gives 1.

2. If graph $G = (V, E)$ is bipartite, then its nodes can be colored with two colors such that the endpoints of each edge get different colors.
   True; bipartite iff 2-colorable

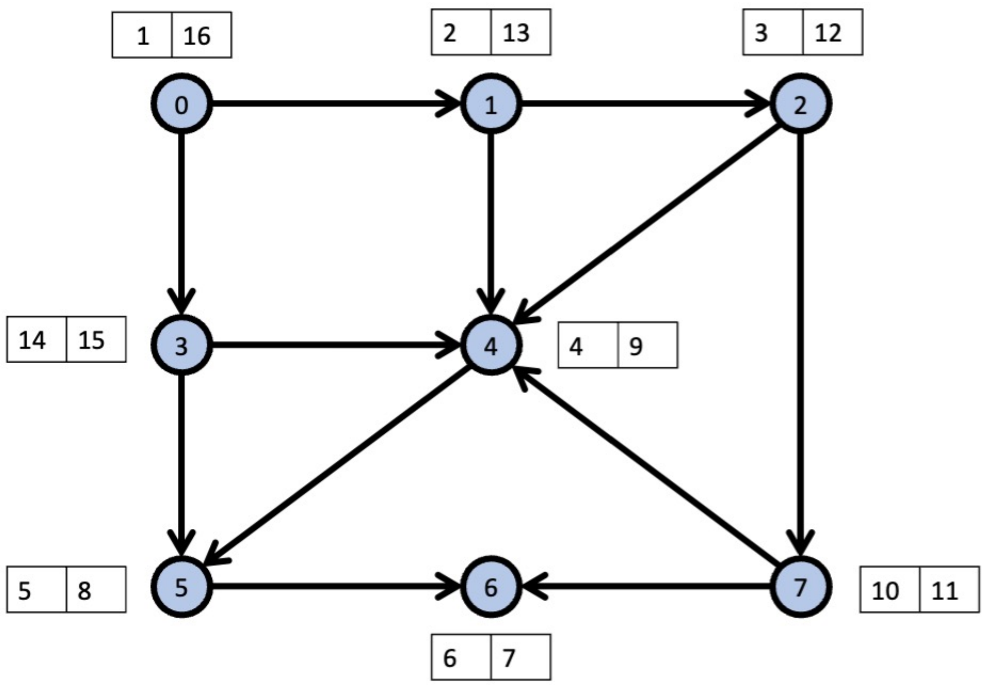
3. If $G = (V, E)$ is a graph with $|V| = n$ vertices, then it must have at least n edges to be a connected graph.
   False; it can be a tree with $n - 1$ nodes. Other counterexamples are also accepted

4. Suppose T is an MST of given weighted $G = (V, E)$, where all edge weights are positive and distinct. If we replace each edge weight, w_e by its square w_e^2 for all $e \in E$, thereby creating a new graph G' with the same vertices but different edge weights, then T must still be an MST of the new graph G'.
   True; if we run the Kruskal's algorithm on the new graph G', it will work in the same way as the edge weights will remain in the same order after sorting. Therefore,the same subset of edges will be selected for the MST of G'.

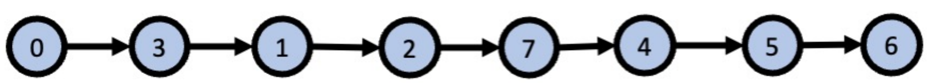
GT Username:	Full Name:

(c) (6 pts) For the following directed graph $G = (V, E)$:

1. (3 pts) Run the DFS algorithm and give the discovery time and finishing time of each node. (In a tie situation, choose nodes in the lexicographical order. For instance, if at some step, you have the option to choose between nodes 2 and 4 for the next vertex to traverse, you must choose vertex 2 first.)



Topology Ordering:



2. (2 pts) Using the result of the DFS traversal obtained in previous part, give a topological ordering for the graph G.
0 - 3 - 1 - 2 - 7 - 4 - 5 - 6

3. (1 pt) Is this graph a directed acyclic graph (DAG)? Explain your answer briefly.
Yes. Because the graph has a topology ordering and we know a directed graph is a DAG if and only if it has a topology ordering.

GT Username:	Full Name:

2 Traversal Applications (10 pts)

The Atlanta Police Department has made all streets one-way in Midtown Atlanta, and states that there is still a way to drive legally from any intersection in the Midtown to any other intersection. A computer program is required to evaluate this statement. Design an efficient (linear) algorithm to accomplish this task by answering the following questions:

(a) (3 pts) Explain how this problem can be formulated as a graph problem and discuss how the corresponding graph can be represented in linear running time.

Graph problem: Consider the Midtown Atlanta as a directed graph $G = (V, E)$ such that the vertices in V represent the intersections in this area, and the directed edges in E represent the streets connecting them. Constructing this graph by the adjacency list representation can be done in $O(|V| + |E|)$

(b) (5 pts) Give an efficient algorithm to evaluate the given statement, and justify the correctness of your solution.

Design: The claim is true if we can get from any vertex to any other vertex, which translates into our graph been strongly connected. Thus, to solve this part we run `SCC` and output YES if G has one strongly connected component and NO otherwise.

Correctness: Let's consider the graph G. To verify the statement, we need to verify if a path from u to v exists for all $u, v \in V$. We know that the above condition will hold true if and only if G is strongly connected.

(c) (2 pts) Discuss the overall running time of your algorithm including the computation required to construct the corresponding graph and the the running time required to evaluate the statement.

Running time Both building the graph and running the `SCC` algorithm take linear time $O(|V| + |E|)$; therefore, the overall running time of the entire algorithm is $O(|V| + |E|)$.

GT Username:	Full Name:

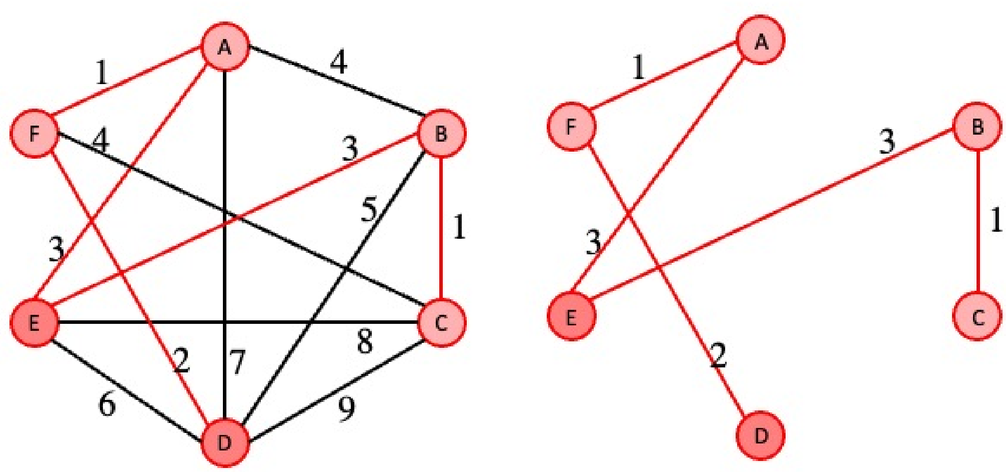
3 Minimum Spanning Tree [12 pts]

Find the minimum spanning tree of the following graph, using Kruskal's and Prim's algorithms. First, briefly explain how, in general, each edge will be selected at each step of these algorithms. Then, in each case, list the steps taken by the algorithm to find the minimum spanning tree, and mark the selected edges on the given graph. (Note, when you list the steps, the order of edges added to the MST matters. In case of a tie, you can choose deliberately.)

(a) (1 pt) Explain how a safe edge is selected to be added to the minimum spanning tree at each step of Kruskal's algorithm.

From lectures, the growing set A is a forest whose vertices are all those of the given graph. The safe edge added to A is always a least-weight edge in the graph that connects two distinct components.

(b) (5 pts) Find the minimum spanning tree using Kruskal's algorithm. Write down the edges of the MST in the order in which they are added to the MST by Kruskal's algorithm.



AF	1
BC	1
FD	2
AE	3
EB	3

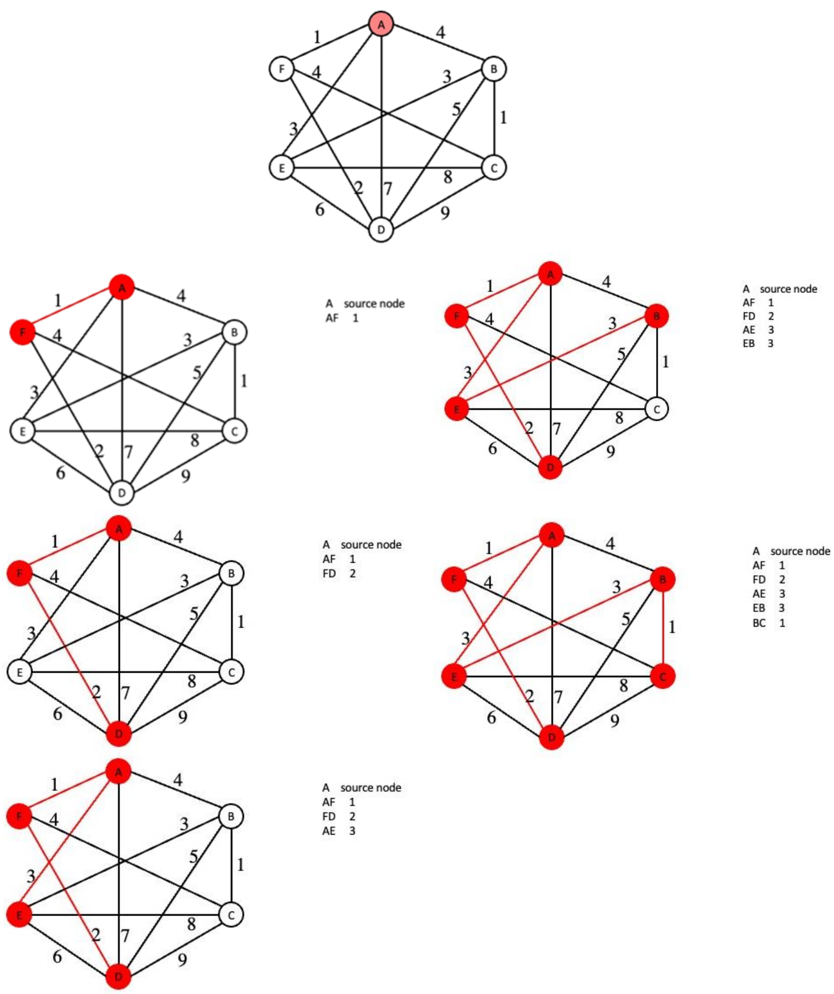
Note we could choose BC before AF. It is also the case for AE and EB, where we could choose EB before AE.

GT Username:	Full Name:

(c) (1 pt) Explain how a safe edge is selected to be added to the minimum spanning tree at each step of Prim's algorithm.

From lectures, the growing set A forms a single tree. The safe edge added to A is always a least-weight edge connecting the tree to a vertex not in the tree.

(d) (5 pts) Find the minimum spanning tree using Prim's algorithm. Write down the edges of the MST in the order in which they are added to the MST by Prim's algorithm.
Assume the algorithm starts at vertex A.
AF $\rightarrow$ FD $\rightarrow$ AE $\rightarrow$ EB$\rightarrow$ BC



GT Username:	Full Name:

4 Minimum Spanning Tree [8 pts]

Give a counter-example or prove the following statement: Let $G = (V, E)$ be a weighted undirected graph. Let C be one cycle in G and let e be an edge in C. If the weight of e is strictly larger than any other edge in C, then e is not in any minimum spanning tree of G.

This statement is **TRUE**.
Proof: We know that a minimum spanning tree cannot have a cycle; therefore, at least one edge of the cycle C cannot be in a minimum spanning tree of G. Let's call this edge e'. For the sake of contradiction, consider $e' \neq e$. we know that the weight of e is strictly larger than any other edge in C, so $e' < e$. Thus, we can swap e' for e in our spanning tree which gives a spanning tree with a smaller weight than the minimum spanning tree. This is contradiction. Therefore, the heaviest weight edge e cannot be in any minimum spanning tree of G. ■

We can also look this problem in this way: Again, let's assume e is in the spanning tree T. If we remove e from T, we have two sub-trees T_1 and T_2. We know that e is an edge in cycle C; therefore, there must be at least another edge (like e') in the cut set of (T_1, T_2) in graph G that can connect T_1 and T_2. Using this edge e', we can define another spanning tree $T' = T - e + e'$ which has a lighter weight than T since e is strictly heavier than e'. Therefore, e is not in any minimum spanning tree of G. ■