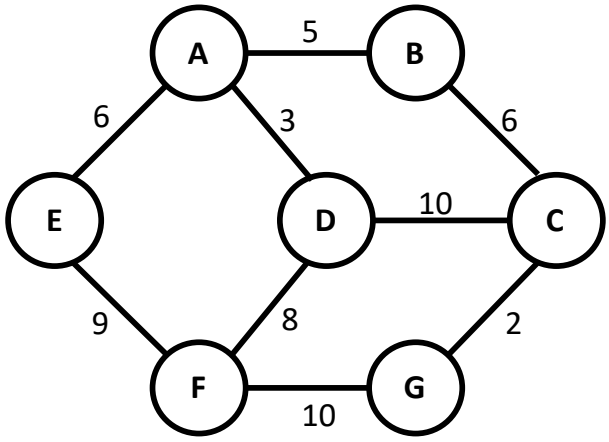
CS-102: Data Structures & Algorithms
Lab 13

Habib University

Objectives: In this lab, we will implement Minimum Spanning Tree algorithms and apply them to solve real-world problems.

Exercise # 1:

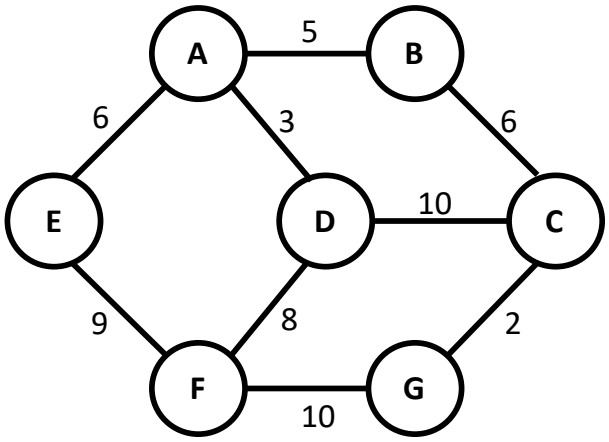
(a) Write a function named "MSTPrims(G, SV)" and implement Prim's algorithm to determine the Minimum Spanning Tree in the following graph.

Example 1:	
Input	G: Adjacency list of the graph shown below  SV: A Note: (you can choose any vertex as a starting vertex) Note: You are free to pass as many arguments as needed to the function "MSTPrims"

(b) Modify your algorithm to find the *Maximum* Spanning Tree.

Exercise # 2:

a) Write a function named "MSTKruskals(EL, N)" and implement Kruskal's algorithm to determine the Minimum Spanning Tree in the following graph.

Example 1 :	
Input	EL: Edge list of the following graph.  Note: You are free to pass as many arguments as needed to the function "MSTKruskals"
Output	`[('C', 'G', 2), ('A', 'D', 3), ('A', 'B', 5), ('A', 'E', 6), ('B', 'C', 6), ('D', 'F', 8)]`

Exercise # 3:

There are eight small islands in a lake, and the state wants to build seven bridges to connect them so that each island can be reached from any other one via one or more bridges. The cost of constructing a bridge is proportional to its length. The distances between pairs of islands are given in the following table.

Use the graph algorithms that you have implemented so far to find which bridges to build to minimize the total construction cost.

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	120
2	-	-	265	175	215	180	185	155
3	-	-	-	260	115	350	435	195
4	-	-	-	-	160	330	295	230
5	-	-	-	-	-	360	400	170
6	-	-	-	-	-	-	175	205
7	-	-	-	-	-	-	-	305
8	-	-	-	-	-	-	-	-

Example 1 :	
Input	Graph: Adjacency list of a graph shown in the table above.
Output	Minimum Spanning Tree covering all of the above nodes.