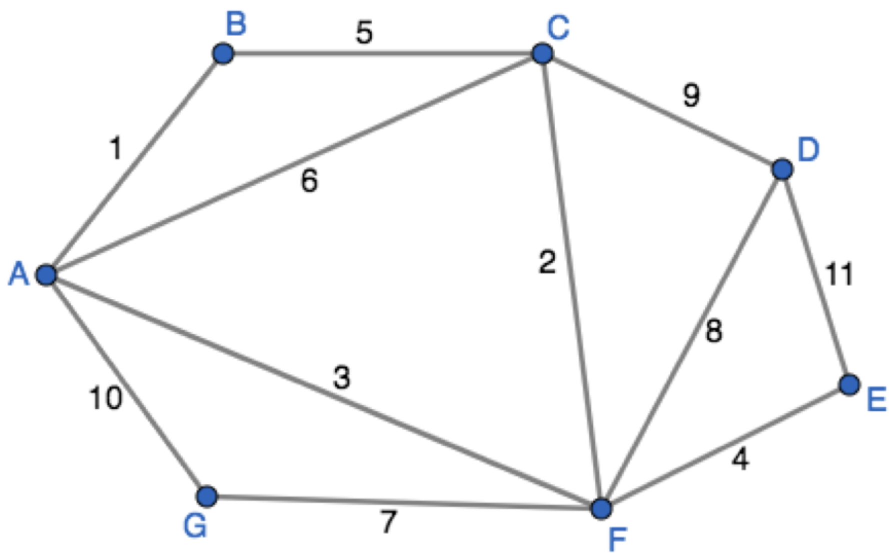
ECS 122A – Algorithm & Analysis
Homework 07 Solution

Question 1 (30 points)

Given the following graph,

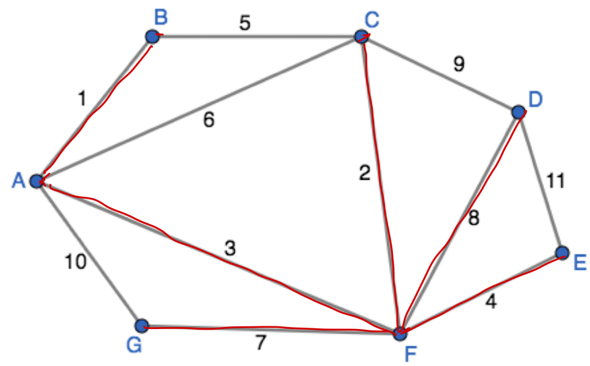


- Find the MST of the graph using Prim's algorithm. State at each step how the key and predecessor arrays are updated.

  Answer:

  The solution here uses A as the original vertex in the inputs.

	A	B	C	D	E	F	G
init	0, N	∞, N	∞, N	∞, N	∞, N	∞, N	∞, N
A		1, A	6, A			3, A	10, A
B			5, B				
F			2, F	8, F	4, F		7, F
C							
E							
G							
D							



- Find the MST of the graph using Kruskal's algorithm. State at each step which edge is added to the forest and which sets are unioned (similar to the table we drew in class).

  Answer:

	A	B	C	D	E	F	G
init	A	B	C	D	E	F	G
(A, B)		A					
(C, F)						C	
(A, F)			A			A	
(E, F)	E	E	E			E	
(B, C)							
(A, C)							
(F, G)							E
(D, F)	D	D	D		D	D	D
(C, D)							
(A, G)							
(D, E)							

Question 2 (20 points)

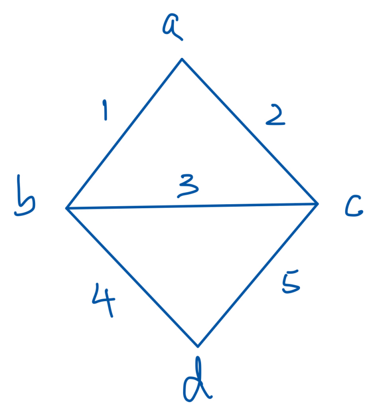
Let $G = (V, E, w)$ be an arbitrary undirected, connected and weighted graph. (Assume the edge weights are distinct.)

Prove or disprove: The minimum spanning tree of G includes the minimum-weight edge in *every* cycle of G.

Answer:

Disprove:

In the following graph, the MST is $\{(a, b), (a, c), (b, d)\}$ but the minimum weight edge of the cycle $\{b, c, d, b\}$ is (b, c).



Question 3 (30 points)

Describe an algorithm that computes the *maximum*-weight spanning of an undirected, connected and weighted graph.

Answer: Negate the edge weights. Run any of the minimum-spanning tree algorithm. The result tree would be the maximum-spanning tree. To get the weight of the tree, simply add up the edge weights and negate it.

Question 4 (20 points)

Given an undirected, connected and weighted graph $G = (V, E, w)$ in which the weight for every edge is 1, describe an algorithm with runtime $O(E)$ that finds the minimum-spanning tree of the graph.

Answer:

Run DFS on the graph. The resulting DFS tree would be a minimum spanning tree. The runtime of DFS is $O(|V| + |E|)$. Since the graph is connected, $|V|$ is $O(|E|)$. So the runtime is $O(|E|)$.