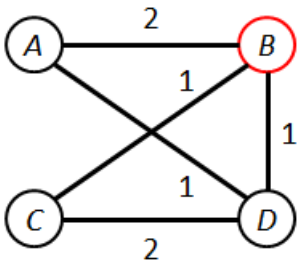
CMSC 451 Homework Assignment 5

Section 8.2 Prim's Minimum Spanning Tree Algorithm

8.1 Give a connected, weighted, undirected graph and a start vertex such that neither the depth-first search tree nor the breadth-first search tree is an MST, regardless of how the adjacency lists are ordered.

Starting at vertex *B*:



8.4 Prove that if the weights on the edges of a connected, undirected graph are distinct, then there is a unique minimum spanning tree.

For a graph G, there exist an MST, T_1. Assume T_1 is not unique and there exists another MST, T_2, with a weight equal to T_1. Since T_1 and T_2 are distinct, they must contain different edges and edge weights since all edge weights are distinct. If there is an edge, e_1, which is in T_1 but not T_2, T_2 must have an edge, e_2, which is not in T_1. If one of the edges in G is decreased by a small amount so as not to change the structure and ordering of T_1, the result is that T_2 will have a smaller weight than before. Therefore T_1 is not minimal and by contradiction, the assumption that T_2 exists is false.

8.6 Execute Prim's minimum spanning tree algorithm by hand on the graph in Figure 8.4(a) showing how the data structures evolve. Clearly indicate which edges become part of the minimum spanning tree and in what order.

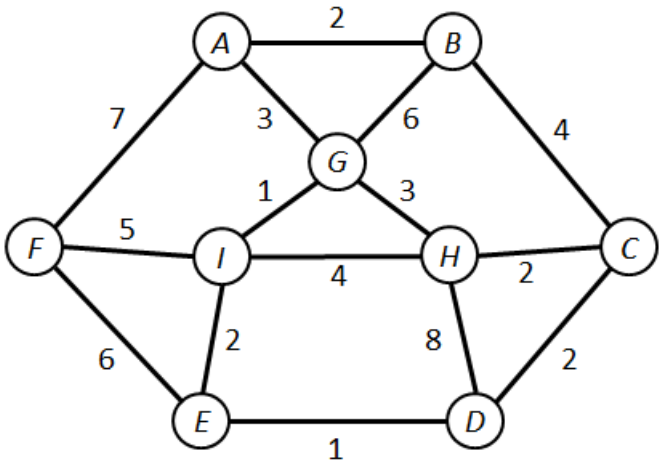
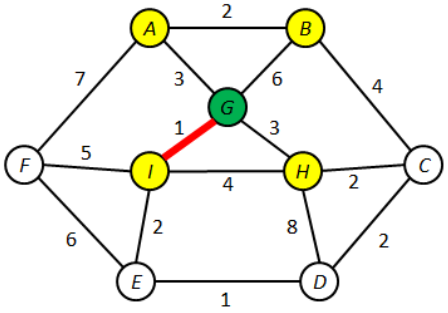


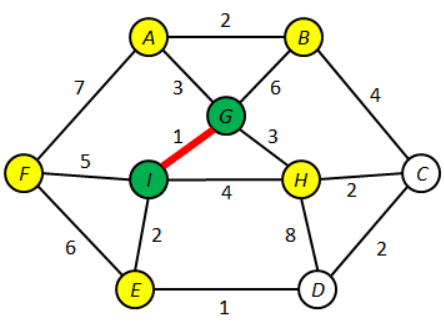
Figure 8.4(a) A weighted graph

a. Start at vertex *G*.



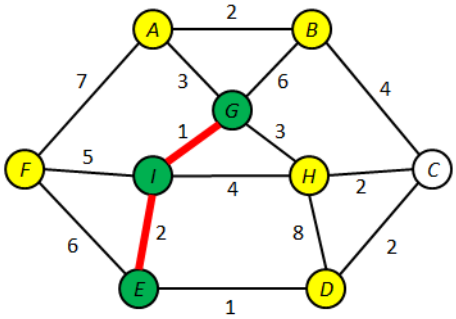
A	3G
B	6G
C	
D	
E	
F	
G	
H	3G
I	1G

Step 1. Choose *G* and place *A*,*B*,*H*, and *I* into fringe.



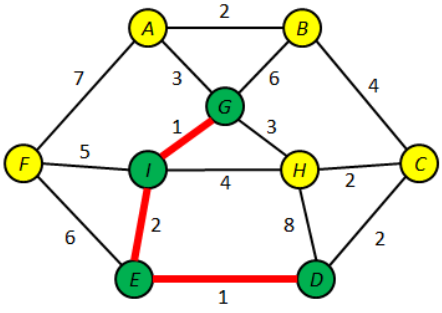
A	3G
B	6G
C	
D	
E	2I
F	5I
G	
H	4I
I	

Step 2. Choose *I* and place *E* into fringe.



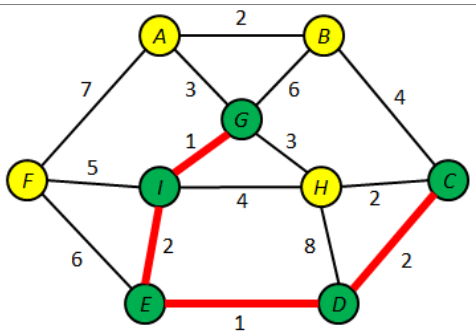
A	3G
B	6G
C	
D	1E
E	
F	6E
G	
H	4I
I	

Step 3. Choose *E* and place *D* into fringe.



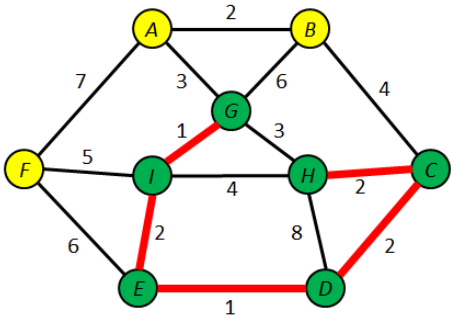
A	3G
B	6G
C	2D
D	
E	
F	5I
G	
H	8D
I	

Step 4. Choose *D* and place *C* into fringe.



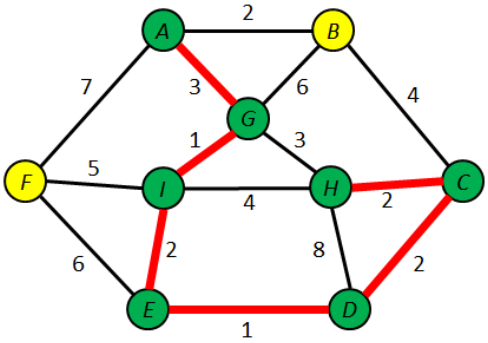
A	3G
B	4C
C	
D	
E	
F	5I
G	
H	2C
I	

Step 5. Choose *C* and place *H* into fringe.



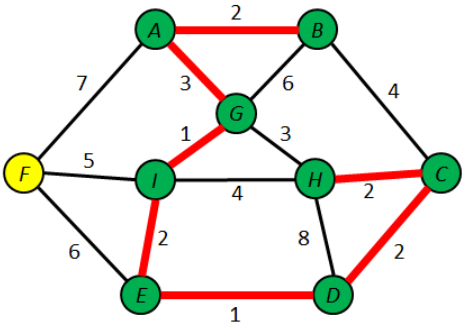
A	3G
B	4C
C	
D	
E	
F	5I
G	
H	
I	

Step 6. Choose *H*.



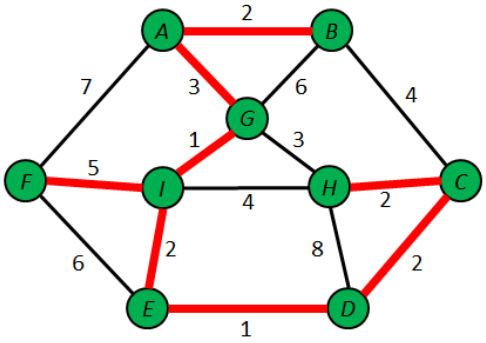
A	
B	2A
C	
D	
E	
F	5I
G	
H	
I	

Step 7. Choose *A*.



A	
B	
C	
D	
E	
F	5I
G	
H	
I	

Step 8. Choose *B*.



A	
B	
C	
D	
E	
F	
G	
H	
I	

Step 9. Choose *F*.

Section 8.4 Kruskal's Minimum Spanning Tree Algorithm

8.24 Find the minimum spanning tree for the graph in Figure 8.14 that would be output by Kruskal's algorithm (Algorithm 8.3), assuming the edges are sorted as shown.

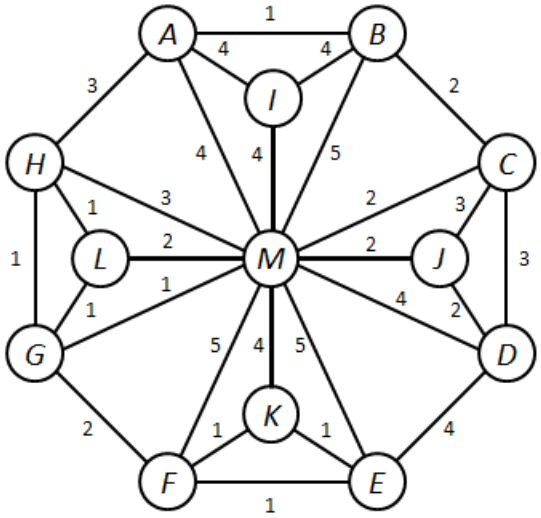
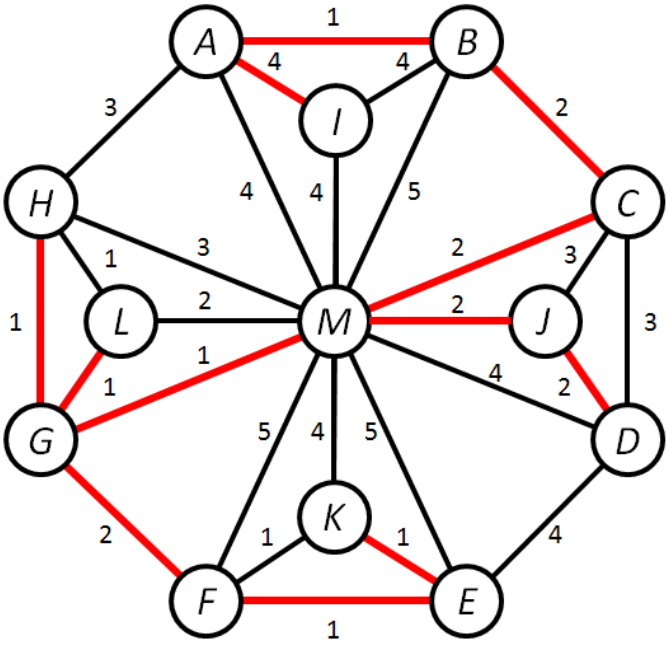


Figure 8.14 Sorted edges: *AB, EF, EK, FK, GH, GL, GM, HL, BC, CM, DJ, FG, JM, LM, AH, CD, CJ, HM, AI, AM, BI, DE, DM, IM, KM, BM, EM, FM*.

AB
EF
EK
~~*FK*~~
GH
GL
GM
~~*HL*~~
BC
CM
DJ
FG
JM
~~*LM*~~
~~*AH*~~
~~*CD*~~
~~*CJ*~~
~~*HM*~~
AI
~~*AM*~~
~~*BI*~~
~~*DE*~~
~~*DM*~~
~~*IM*~~
~~*KM*~~
~~*BM*~~
~~*EM*~~
~~*FM*~~



ABCGHLMEFKDJI

Action	Disjoint union
Take *AB*	*AB*
Take *EF*	*AB, EF*
Take *EK*	*AB, EFK*
Drop *FK*	
Take *GH*	*AB, EFK, GH*
Take *GL*	*AB, EFK, GHL*
Take *GM*	*AB, EFK, GHLM*
Drop *HL*	
Take *BC*	*ABC, EFK, GHLM*
Take *CM*	*ABCGHLM, EFK*
Take *DJ*	*ABCGHLM, EFK, DJ*
Take *FG*	*ABCGHLMEFK, DJ*
Take *JM*	*ABCGHLMEFKDJ*
Drop *LM, AH, CD,CJ, HM*	
Take *AI*	*ABCGHLMEFKDJI*
Drop all remaining	