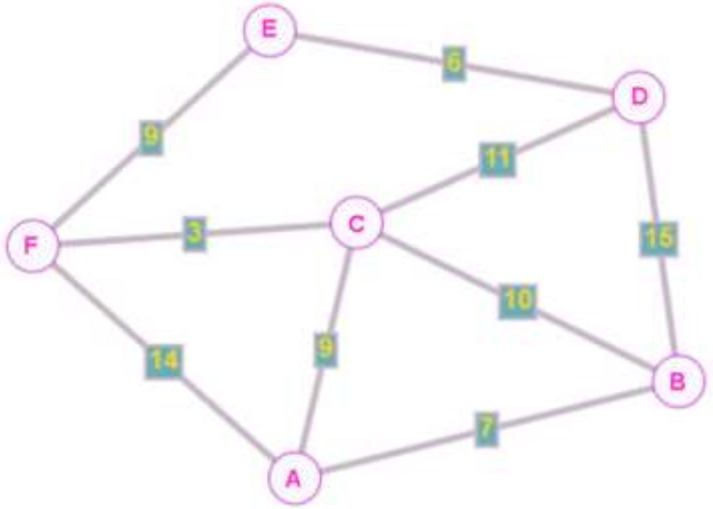
OLQ8

Practice and Review

Look over the these examples of how to answer the Prim's Algorithm questions and Kruskal's Algorithm for OLQ8.

Use **Prim**'s Algorithm to find the MST starting at Vertex A.

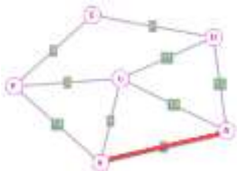


Prim's would start at Vertex A and create the following list and order it

(A,B,7)

(A,C,9)

(A,F,14)



Prim's would pick (A,B,7) since 7 is the smallest weight and adding this edge does not create a cycle.

Remove (A,B,7) from the list.

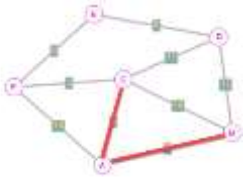
Move to Vertex B and add edges incident to B to the list in order of weight that have not already been added.

(A,C,9)

(B,C,10)

(A,F,14)

(B,D,15)



Prim's would pick (A,C,9) since 9 is the smallest weight and adding this edge does not create a cycle.

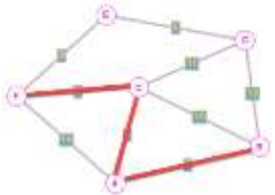
Remove (A,C,9) from the list.

Move to Vertex C and add edges incident to C to the list in order of weight that have not already been added.

(C,F,3)

(B,C,10)

(C,D,11)



(A,F,14)

(B,D,15)

Prim’s would pick (C,F,3) since 3 is the smallest weight and adding this edge does not create a cycle.

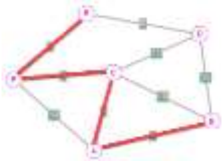
Remove (C,F,3) from the list.

Move to Vertex F and add edges incident to F to the list in order of weight that have not already been added.

(F,E,9)

(B,C,10)

(C,D,11)



(A,F,14) – did not add (F,A,14) because the graph is undirected and those two are equivalent.

(B,D,15)

Prim’s would pick (F,E,9) since 9 is the smallest weight and adding this edge does not create a cycle.

Remove (F,E,9) from the list.

Move to Vertex E and add edges incident to E to the list in order of weight that have not already been added.

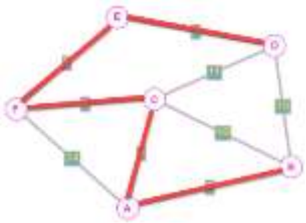
(E,D,6)

(B,C,10)

(C,D,11)

(A,F,14)

(B,D,15)

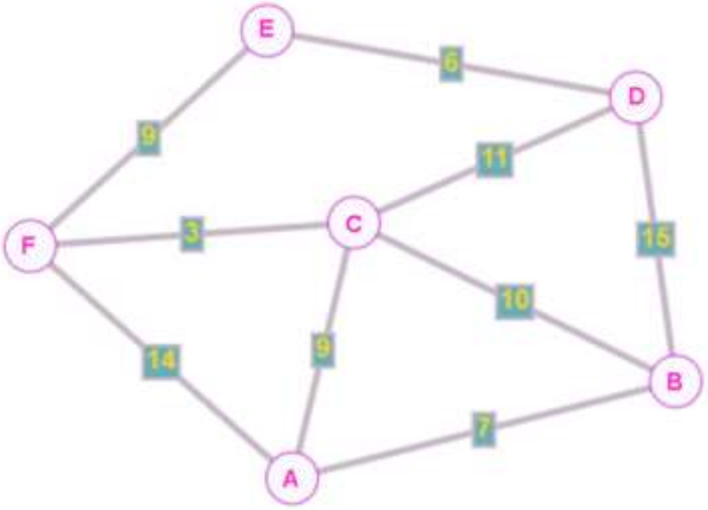


Prim’s would pick (E,D,6) since 6 is the smallest weight and add this edge does not create a cycle.

Remove (E,D,6) from the list.

The algorithm could be stopped at this point by either recognizing that all vertices are now in the MST or it could stop by recognizing that the number of edges added to the MST is |*V*| -1 or it could continue processing the rest of the list and recognize that all of the rest of the edges will cause cycles and will be discarded.

Use **Kruskal**'s Algorithm to find the MST.



Kruskal's would start by creating an ordered list of all edges.

(C,F,3)

(E,D,6)

(A,B,7)

(A,C,9)

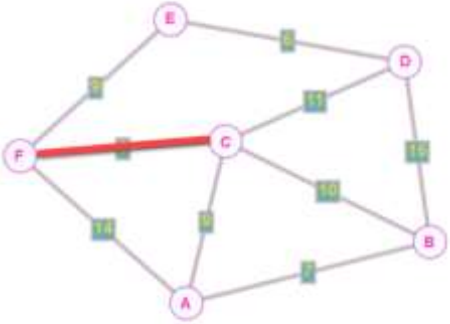
(F,E,9)

(B,C,10)

(C,D,11)

(A,F,14)

(B,D,15)



Kruskal's would pick (C,F,3) since 3 is the smallest weight and adding this edge does not create a cycle.

Remove (C,F,3) from the list.

(E,D,6)

(A,B,7)

(A,C,9)

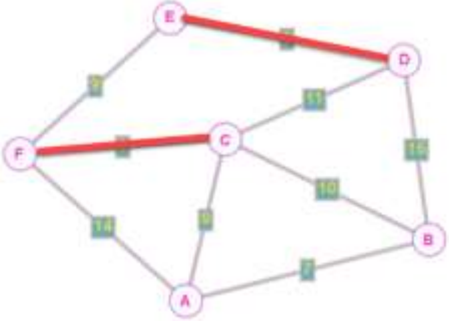
(F,E,9)

(B,C,10)

(C,D,11)

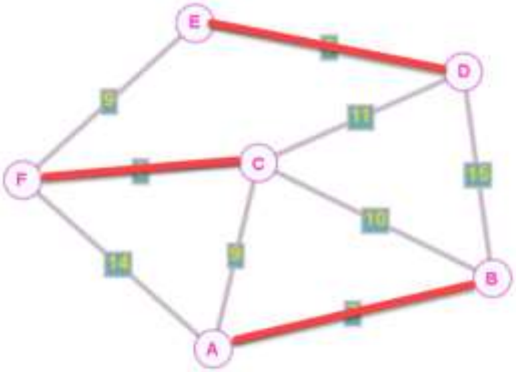
(A,F,14)

(B,D,15)



Kruskal's would pick (E,D,6) since 6 is the smallest weight and adding this edge does not create a cycle

Remove (E,D,6) from the list.



(A,B,7)

(A,C,9)

(F,E,9)

(B,C,10)

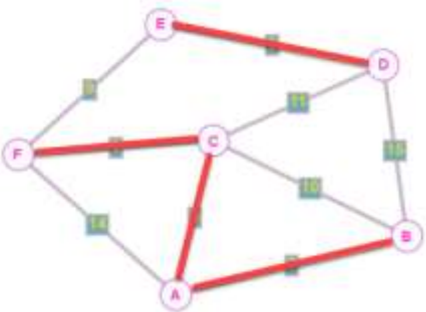
(C,D,11)

(A,F,14)

(B,D,15)

Kruskal’s would pick (A,B,7) since 7 is the smallest weight and adding this edge does not create a cycle.

Remove (A,B,7) from the list.



(A,C,9)

(F,E,9)

(B,C,10)

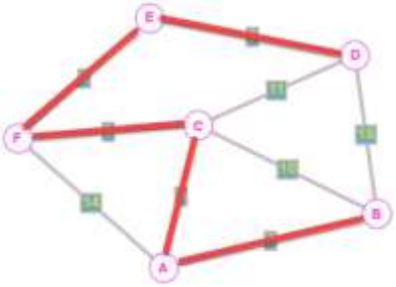
(C,D,11)

(A,F,14)

(B,D,15)

Kruskal’s would pick (A,C,9) since 9 is the smallest weight and adding this edge does not create a cycle.

Remove (A,C,9) from the list.



(F,E,9)

(B,C,10)

(C,D,11)

(A,F,14)

(B,D,15)

Kruskal’s would pick (F,E,9) since 9 is the smallest weight and add this edge does not create a cycle.

Remove (F,E,9) from the list.

The algorithm could be stopped at this point by either recognizing that all vertices are now in the MST or it could stop by recognizing that the number of edges added to the MST is |*V*| -1 or it could continue processing the rest of the list and recognize that all of the rest of the edges will cause cycles and will be discarded.