Minimum Spanning Trees

CSCI 170 Spring 2021

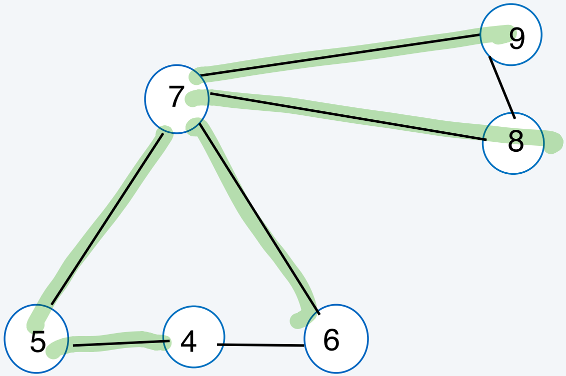
Sandra Batista

1.1-1.2

Spanning Trees

In an undirected connected graph G, **a spanning tree** is a subgraph that is a tree and that contains all vertices of G.

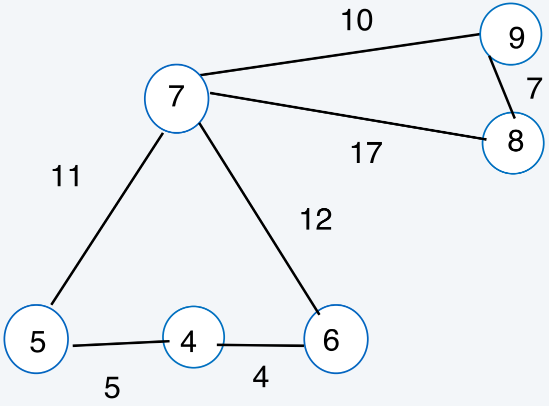
DFS and BFS on connected graphs both produce spanning trees of the graphs.



Minimum Spanning Trees

Now let’s consider if we put weights on the edges. The edge weights can represent part of a problem we want to model such as distances or costs.

For BFS and DFS we did not consider edge weights.



Example for a shipping company:
Let vertices can be the factories,
edges connect factories if they are on a shipping route,
and the edge weight is the distance between the factories.

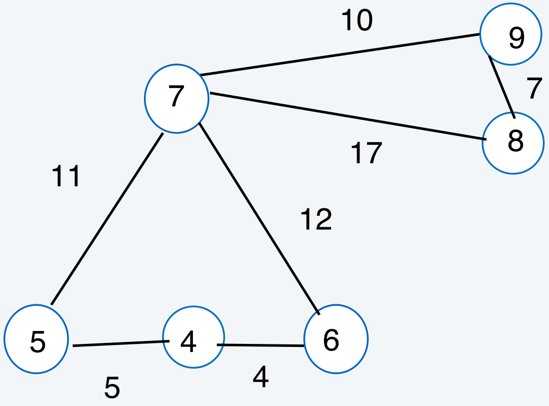
How can the shipping company visit all the factories, but in the least distance traveled?

Minimum Spanning Trees

Given a connected undirected graph, G= (V,E) with positive edge weights, find the spanning tree with the minimum sum of the edge weights. This is the **minimum spanning tree (MST).**

The sum of the edge weights is the **cost of the MST.**

Would a greedy algorithm work for this problem?

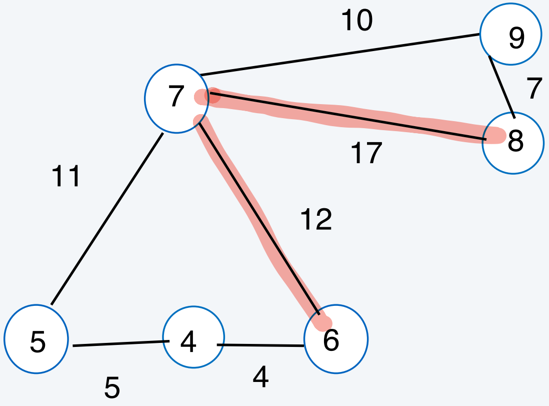


Example for a shipping company:
Let vertices can be the factories,
edges connect factories if they are on a shipping route,
and the edge weight is the distance between the factories.

How can the shipping company visit all the factories, but in the least distance traveled?

Cycle Property of MST

The **cycle property of MST**: for any cycle in the graph, if e is the maximum cost edge in the cycle, then e is not in the MST.

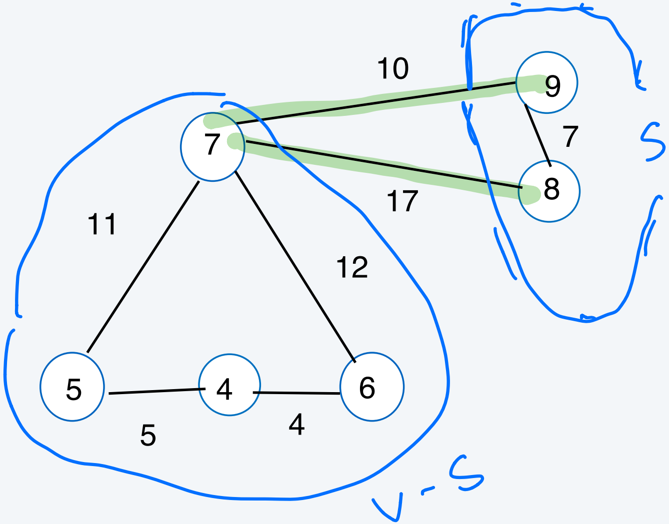


What edges can never be in the MST for this graph?

The cut of a graph

A **cut** is a partition of the vertices of the graph into two sets: S and V-S. These sets do not intersect but are the entire set of vertices.

An edge, e = (u,v) **spans** the cut if u is in S and v is in V-S.

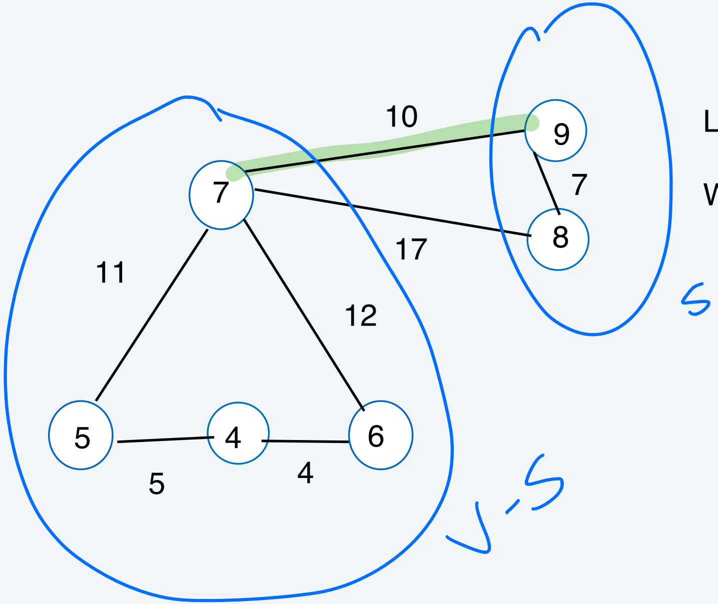


Let a cut in this graph be S = {8,9} and V-S = {4,5,6,7}.

What edges span the cut?

Cut Property of MST

The **cut property of MST**: for any subset of vertices S, if e is the minimum cost edge between S and V-S, then e is in the MST.



Let a cut in this graph be S = {8,9} and V-S = {4,5,6,7}.

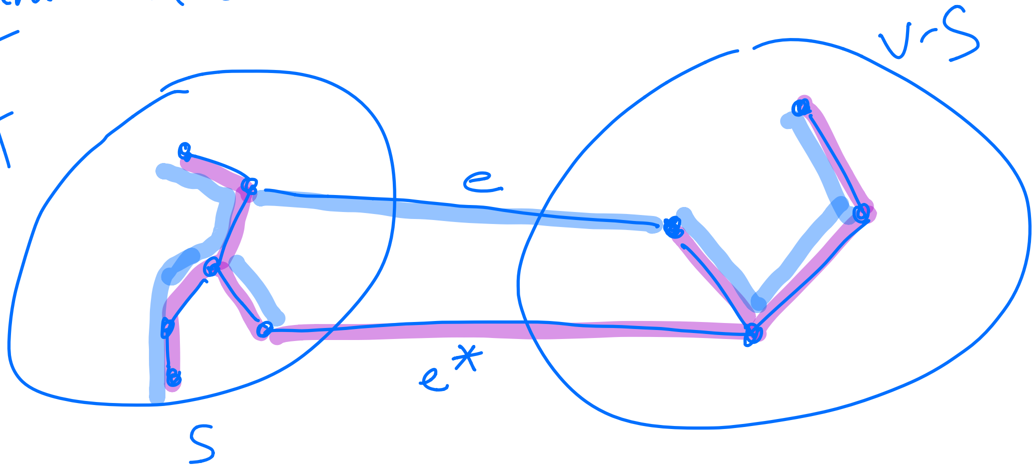
What edge must be in the MST?

Exercise: Cut Property of MST Proof

Prove the following by contradiction: For any subset of vertices S, if e is the minimum cost edge between S and V-S, then e is in the MST.

Proof by contradiction: Assume $\exists$ $S \subseteq V$ $e = (u, v)$ $u \in S$ and $v \in V\text{-}S$ of minimum cost and e is not in the MST.

Let T be MST



$\text{cost}(e) < \text{cost}(e^*)$

Exercise: Cut Property of MST Proof

Prove the following by contradiction: For any subset of vertices S, if e is the minimum cost edge between S and V-S, then e is in the MST.

Proof by contradiction (continued):

Let T' be spanning tree st

$$T' = T - e^* \cup e$$

Since $cost(e) < cost(e^*)$

$$cost(T') < cost(T)$$

$\Rightarrow \Leftarrow$ T MST

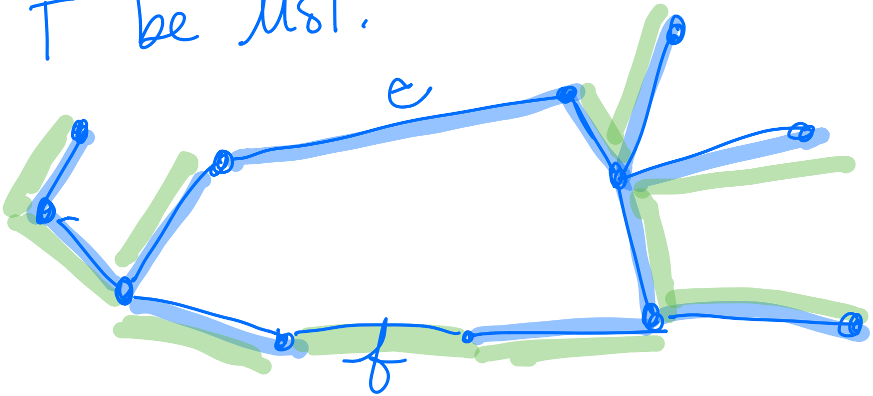
Exercise: Cycle Property of MST Proof

Prove the following by contradiction: For any cycle in the graph, if e is the maximum cost edge in the cycle, then e is not in the MST.

Proof by contradiction: Assume ∃ cycle in graph e is the maximum cost edge in the cycle and e is in MST

Let T be MST.

cost(f) < cost(e)



Exercise: Cycle Property of MST Proof

Prove the following by contradiction: For any cycle in the graph, if e is the maximum cost edge in the cycle, then e is not in the MST.

Proof by contradiction (continued):

$\exists$ edge f in cycle st cost(f) < cost(e)

construct spanning tree $T' = T - e \cup f$

$\Rightarrow$ T' is spanning tree

$\Rightarrow$ since cost(f) < cost(e) $\Rightarrow$ cost(T') < cost(T)

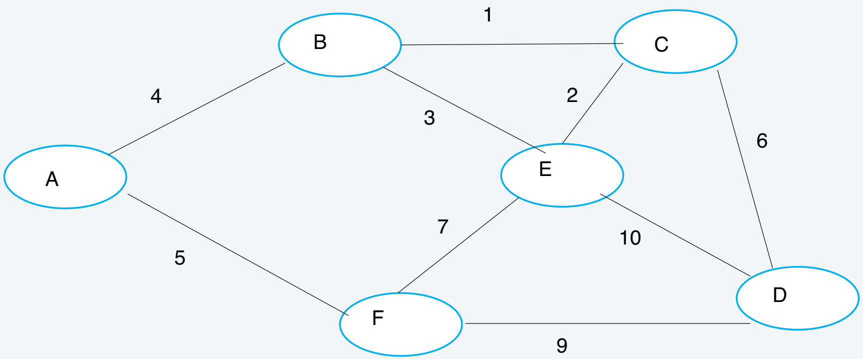
$\Rightarrow\!\!\!\!\!/ \Leftarrow$ (T is MST)

Kruskal’s Algorithm

Given a connected graph, G=(V,E), with positive weights on E

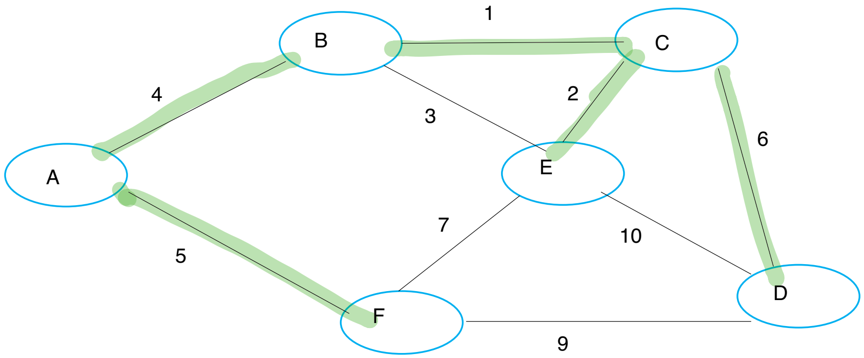
1. Sort the edge weights of E.
2. At each iteration, add the minimum cost edge to T* that does not create a cycle.
3. Stop when all vertices are included in T*

Claim: Kruskal’s algorithm produces T*, an MST for G.



Exercise: Trace of Kruskal's algorithm on the shipping company graph

Iteration:	Vertices of T* = {}	Edges of T* = {}
0		
1	{B, C}	(B, C)
2	{B, C, E}	(C, E)
3	{B, C, E, A}	(A, B)
4	{B, C, E, A, F}	(A, F)
5	{B, C, E, A, F, D}	(C, D)

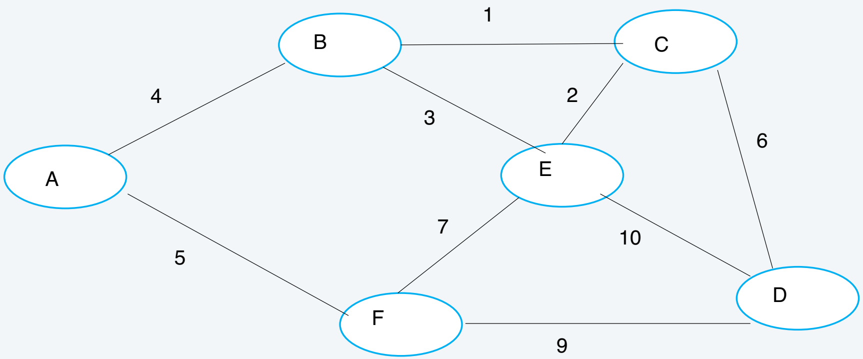


Prim's Algorithm

Given a connected graph, G=(V,E), with positive weights on E

1. Choose any arbitrary vertex, s, to be the starting vertex in the tree, T*.
2. At each iteration, add the lowest cost edge connecting vertex in T* to vertex not in T*. (The cut is the vertices in T* and those vertices not in T* yet.)
3. Stop when all vertices are included in T*

Claim: Prim's algorithm produces T*, an MST for G.



Exercise: Trace of Prim's algorithm on the shipping company graph

Iteration: 0

Vertices of T* = {A}

Edges of T* = {}

Iteration	Vertices of T*	Edges of T*
0	{A}	{}
1	{A, B}	(A, B)
2	{A, B, C}	(B, C)
3	{A, B, C, E}	(C, E)
4	{A, B, C, E, F}	(A, F)
5	{A, B, C, E, F, D}	(C, D)