

1 A circuit in which only the start/end vertex occurs twice is called a **cycle**

2 A connected graph without any cycles is called a **tree**.

Theorem A connected graph G is a tree if and only if $v = e + 1$, where v is the number of vertices and e is the number of edges in G .

3 A **spanning tree** for a graph G is a tree T such that:

- (i) every edge of T is an edge of G , and
- (ii) the vertices of T are exactly the same as the vertices of G .

The problem of finding the spanning tree of smallest weight, we need to consider the edges **in order of increasing weight**. One algorithm is known as **Kruskal's Algorithm**. Kruskal's Algorithm also keeps track of the accumulated weights of the edges. The weight of an edge often represents the cost of creating a connection in a network.

Kruskal's Algorithm

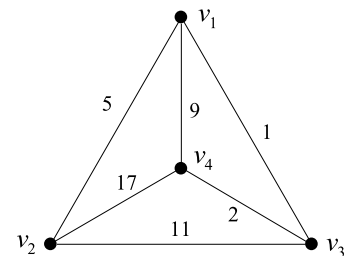
Input: $n, m, V_1, \dots, V_n, e_1, \dots, e_m, w_1, \dots, w_m$

Output: $\{e_{i_j}\}_{j=1}^{n-1}, W$ [The edge set and weight of the spanning tree.]

- (1) Sort the edge set in order of increasing weight.
- (2) For $i = 1$ to n
 - (a) let $N(i) = i$. [This gives the initial labels.]
 - (b) let $\epsilon_i =$ the i^{th} edge in the ordered edge list
- (3) $E \leftarrow \emptyset$. [This gives the initial set of edges.]
- (4) $W \leftarrow 0$. [This gives the initial weight]
- (5) $Count \leftarrow 0$
- (6) For $r = 1$ to m
 - (a) If $\epsilon_r = \{v_i, v_j\}$ and $N(i) \neq N(j)$ then
 - (i) $E \leftarrow E \cup \{\epsilon_r\}$ (ii) $W \leftarrow W + w_r$ (iii) $Count \leftarrow Count + 1$
 - (iv) For $k = 1$ to n (α) if $N(k) = N(i)$ or $N(k) = N(j)$ then $N(k) \leftarrow \min(N(i), N(j))$
 - (v) if $Count = n - 1$ then Output E, W .
 - (b) If $\epsilon_r = \{v_i, v_j\}$ and $N(i) = N(j)$ then do nothing, move to next r .

In Summary

- (1) First order the edges — put them in order of increasing weight
- (2) Construct a table — apply Kruskal's Algorithm
- (3) Write down the conclusion — the edges of the spanning tree & its weight.



Example 1 Edge List: $e_1 = v_2v_4$ $e_2 = v_2v_3$ $e_3 = v_1v_4$ $e_4 = v_1v_2$ $e_5 = v_3v_4$ $e_6 = v_1v_3$

	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1						
After Step 2						
After Step 3						

Spanning Tree 1:
Edges:
Weight:

Example 2 Edge List: $e_1 = v_1v_4$ $e_2 = v_1v_2$ $e_3 = v_2v_3$ $e_4 = v_3v_4$ $e_5 = v_1v_3$ $e_6 = v_2v_4$

	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1						
After Step 2						
After Step 3						

Spanning Tree 2:
Edges:
Weight:

Example 3 Edge List: $e_1 = v_1v_3$ $e_2 = v_3v_4$ $e_3 = v_1v_4$ $e_4 = v_1v_2$ $e_5 = v_2v_3$ $e_6 = v_1v_3$

	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1						
After Step 2						
After Step 3						
After Step 4						

Spanning Tree 3:

Edges:

Weight:

1	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1	1	2	3	2	e_1	17
After Step 2	1	2	2	2	e_1, e_2	28
After Step 3	1	1	1	1	e_1, e_2, e_3	33

Minimal Spanning Tree:

Edges: e_1, e_2, e_3

Weight: 33

2	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1	1	2	3	1	e_1	5
After Step 2	1	1	3	1	e_1, e_2	14
After Step 3	1	1	1	1	e_1, e_2, e_3	25

Minimal Spanning Tree:

Edges: e_1, e_2, e_3

Weight: 25

3	N(1)	N(2)	N(3)	N(4)	Edges	Weight
Initial Values	1	2	3	4	\emptyset	0
After Step 1	1	2	1	4	e_1	1
After Step 2	1	2	1	1	e_1, e_2	3
After Step 3	1	2	1	1	e_1, e_2	3
After Step 4	1	1	1	1	e_1, e_2, e_4	12

Minimal Spanning Tree:

Edges: e_1, e_2, e_4

Weight: 8

Using Kruskal's Algorithm to find a minimal spanning tree for the graph with:

Vertices: v_1 v_2 v_3 v_4 v_5 v_6

Edges: v_1v_2 v_1v_3 v_1v_4 v_1v_5 v_1v_6 v_2v_3 v_2v_6 v_3v_4 v_3v_5 v_4v_5 v_5v_6

Weights: 8 9 7 11 10 6 3 5 1 2 4

(The edges and weights are listed in corresponding order.)

$e_1 = \dots\dots\dots$ $e_2 = \dots\dots\dots$ $e_3 = \dots\dots\dots$ $e_4 = \dots\dots\dots$ $e_5 = \dots\dots\dots$ $e_6 = \dots\dots\dots$

$e_7 = \dots\dots\dots$ $e_8 = \dots\dots\dots$ $e_9 = \dots\dots\dots$ $e_{10} = \dots\dots\dots$ $e_{11} = \dots\dots\dots$

	N(1)	N(2)	N(3)	N(4)	N(5)	N(6)	Edges	W
Initially	1	2	3	4	5	6	\emptyset	0
1.								
2.								
3.								
4.								
5.								
6.								
7.	1	1	1	1	1	1	e_1, e_2, e_3, e_4, e_7	17

Minimal Spanning Tree

Edges: e_1, e_2, e_3, e_4, e_7 or $v_3v_5, v_4v_5, v_2v_6, v_5v_6, v_1v_4$

Weight: 17