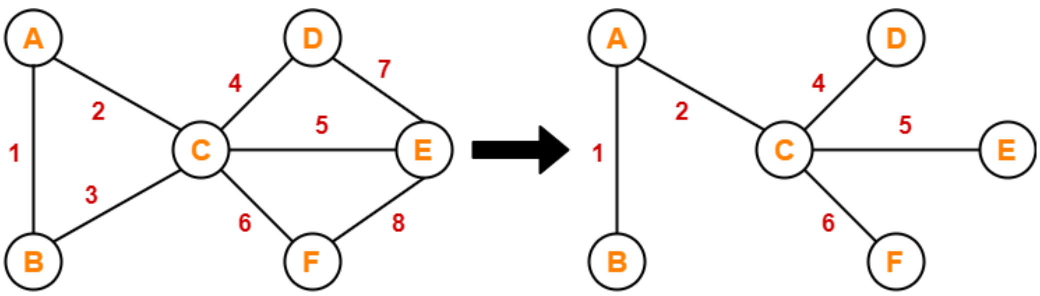
Important comparision

Concept-01:

If all the edge weights are distinct, then both the algorithms are guaranteed to find the same MST.

Example-

Consider the following example-



Given Graph

Minimum Spanning Tree (MST)
(Cost = 18 units)

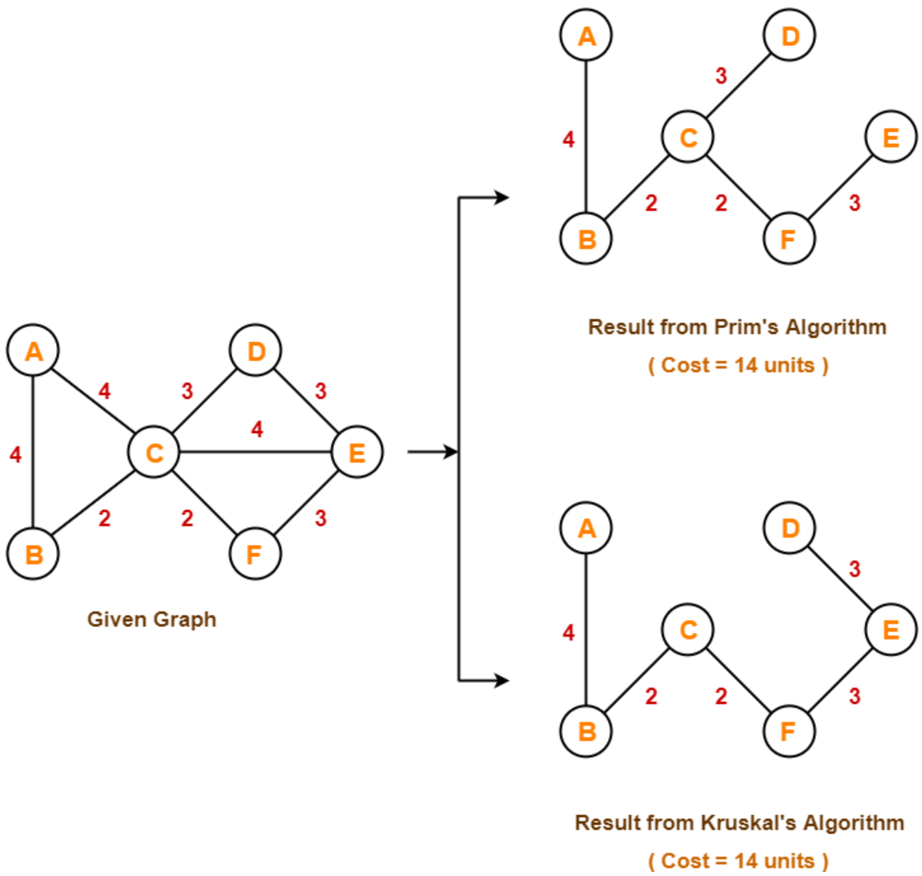
Here, both the algorithms on the above given graph produces the same MST as shown.

Concept-02:

- If all the edge weights are not distinct, then both the algorithms may not always produce the same MST.
- However, cost of both the MST_s would always be same in both the cases.

Example-

Consider the following example-



Here, both the algorithms on the above given graph produces different MST_s as shown but the cost is same in both the cases.

Concept-03:

Kruskal's Algorithm is preferred when-

- The graph is sparse.
- There are less number of edges in the graph like E = O(V)
- The edges are already sorted or can be sorted in linear time.

Prim's Algorithm is preferred when-

- The graph is dense.
- There are large number of edges in the graph like $E = O(V^2)$.

Kruskal's Algorithm	Prim's Algorithm
It starts to build the Minimum Spanning Tree from the vertex carrying minimum weight in the graph.	It starts to build the Minimum Spanning Tree from any vertex in the graph.
It traverses one node only once.	It traverses one node more than one time to get the minimum distance.
Kruskal's algorithm can generate forest(disconnected components) at any instant as well as it can work on disconnected components	Prim's algorithm gives connected component as well as it works only on connected graph.
Kruskal's algorithm runs faster in sparse graphs.	Prim's algorithm runs faster in dense graphs.
Kruskal's algorithm uses Heap Data Structure.	Prim's algorithm uses List Data Structure.