



SNS COLLEGE OF TECHNOLOGY  
(Autonomous )  
COIMBATORE-35



# Minimum Spanning Tree – Prim's and Kruskals Algorithm



# Definition

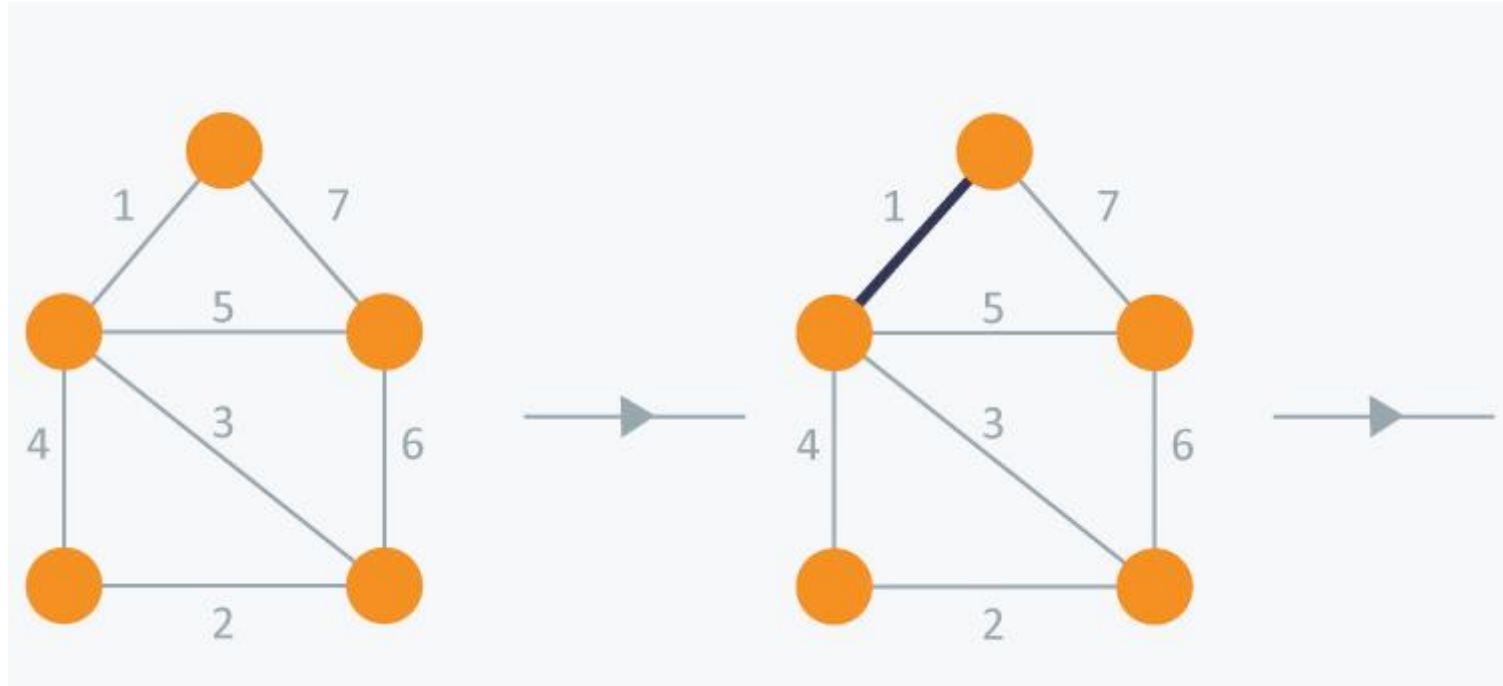
**Kruskal's Algorithm** builds the spanning tree by adding edges one by one into a growing spanning tree. Kruskal's algorithm follows greedy approach as in each iteration it finds an edge which has least weight and add it to the growing spanning tree.

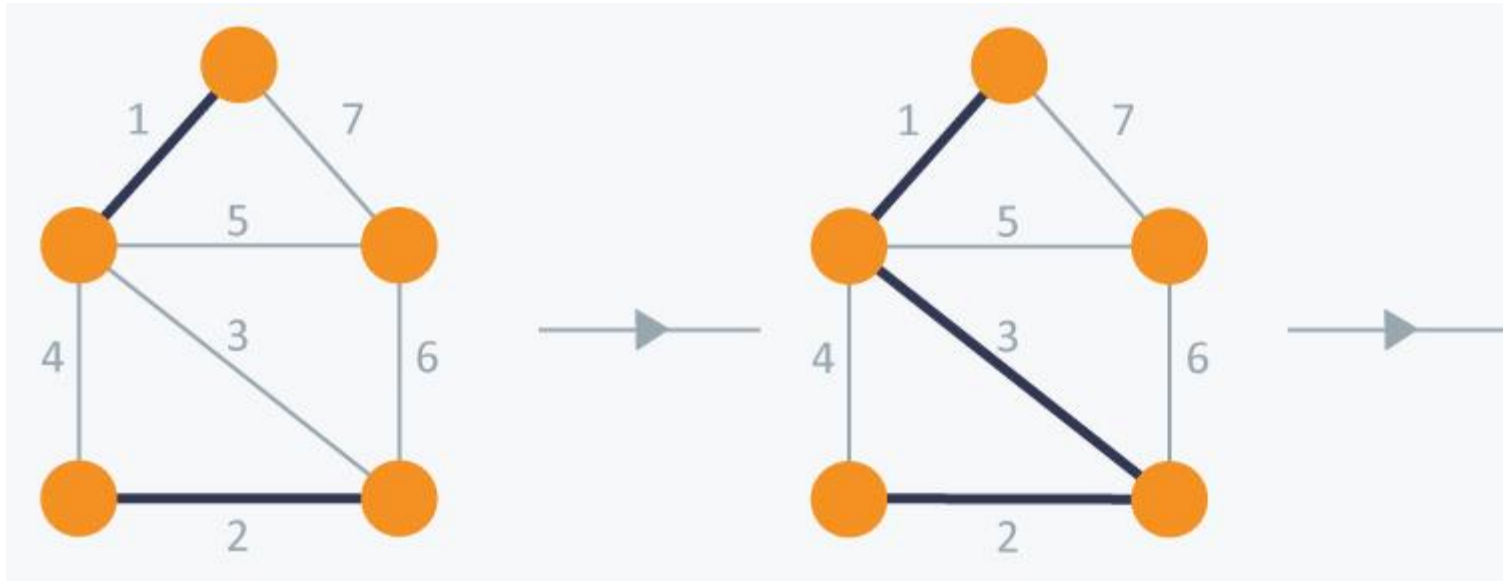
## Algorithm Steps:

Sort the graph edges with respect to their weights.  
Start adding edges to the MST from the edge with the smallest weight until the edge of the largest weight.  
Only add edges which doesn't form a cycle , edges which connect only disconnected components.



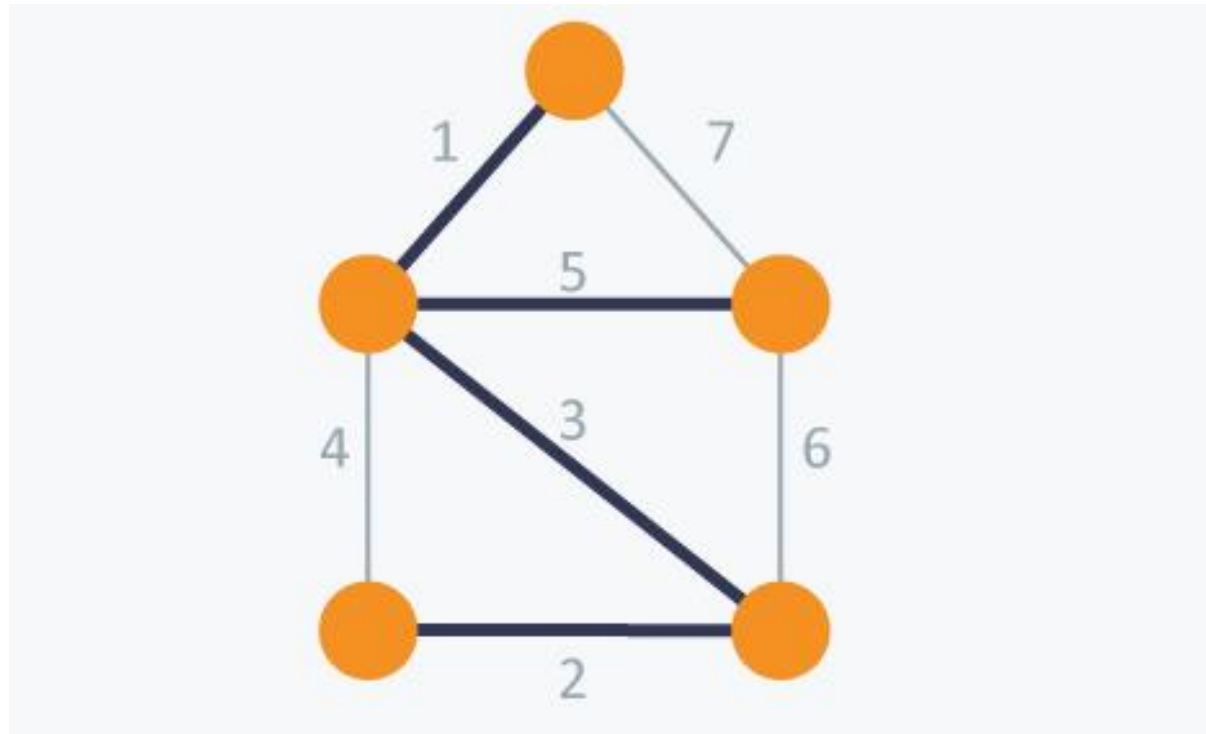
# Example







# Final Tree with minimum cost= 11





# Final Tree with minimum cost= 11

## Prim's Algorithm

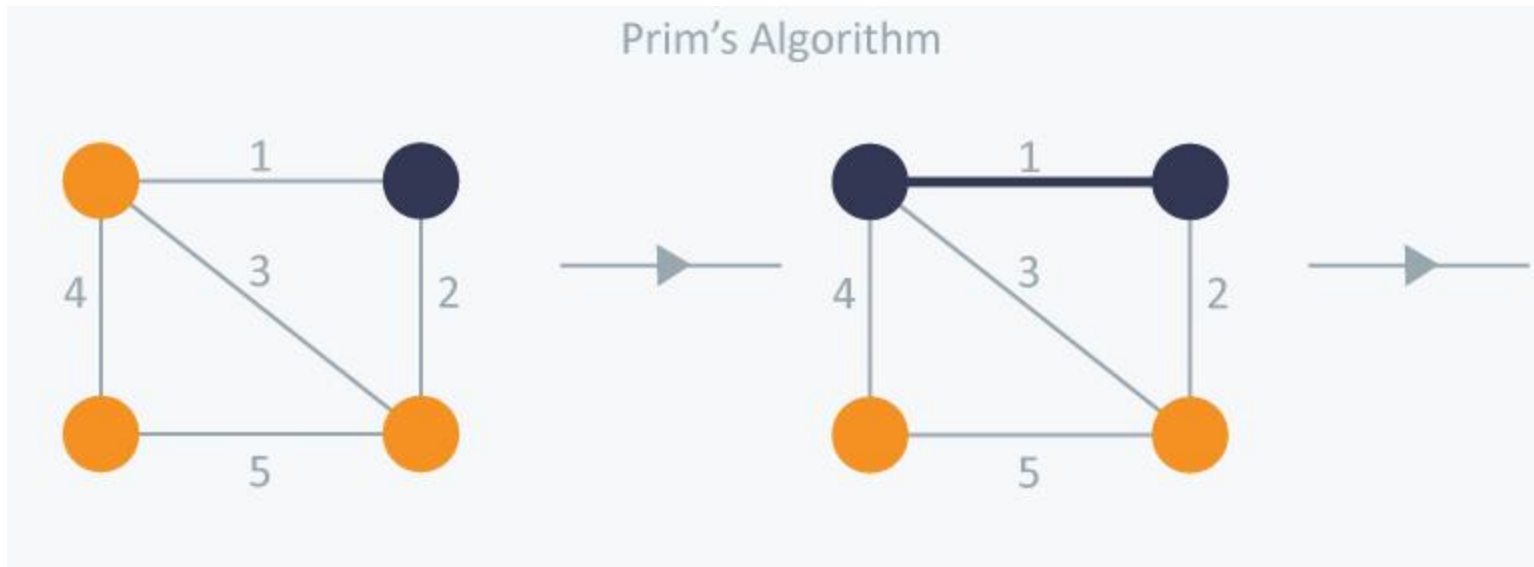
Prim's Algorithm also use Greedy approach to find the minimum spanning tree. In Prim's Algorithm we grow the spanning tree from a starting position. Unlike an **edge** in Kruskal's, we add **vertex** to the growing spanning tree in Prim's.

## Algorithm Steps:

Maintain two disjoint sets of vertices. One containing vertices that are in the growing spanning tree and other that are not in the growing spanning tree. Select the cheapest vertex that is connected to the growing spanning tree and is not in the growing spanning tree and add it into the growing spanning tree. This can be done using Priority Queues. Insert the vertices, that are connected to growing spanning tree, into the Priority Queue. Check for cycles. To do that, mark the nodes which have been already selected and insert only those nodes in the Priority Queue that are not marked.



# Example





# Example – Minimum cost- 7

