Day - 2

Introduction to algorithms

Made by -
Shivam Joshi

Minimum spanning tree

We will be discussing about MST using two algorithms-

1. Kruskal’s algorithm
2. Prim’s algorithm

Kruskal's algorithm

Below are the steps for finding MST using Kruskal's algorithm -

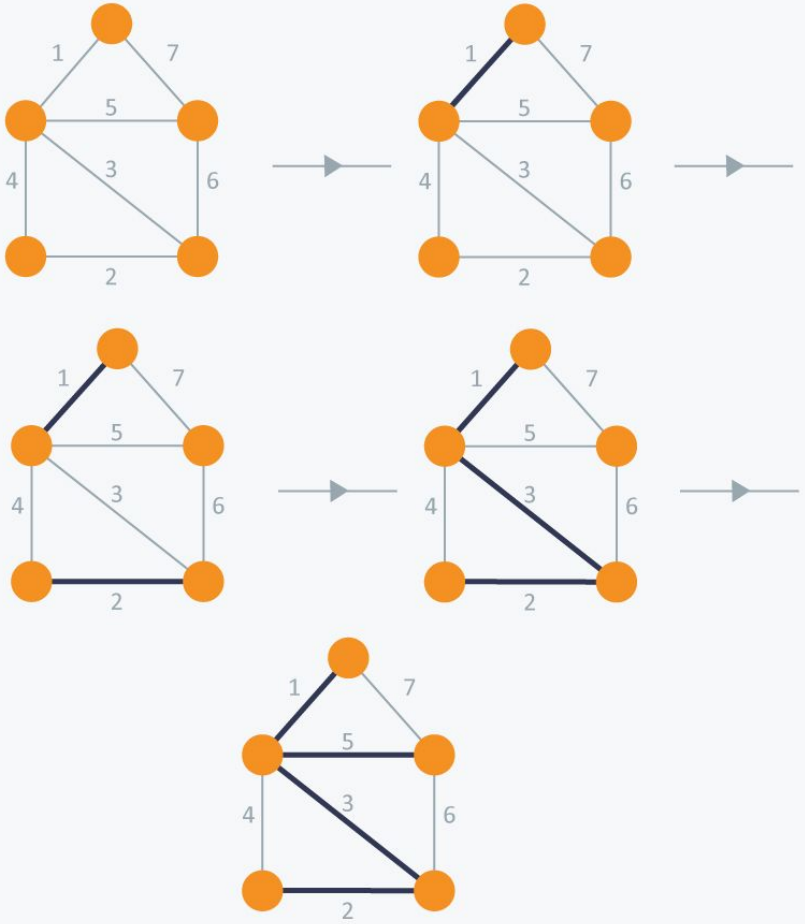
1. Sort all the edges in non-decreasing order of their weight.

2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.

3. Repeat step#2 until there are (V-1) edges in the spanning tree.

The algorithm is a Greedy Algorithm. The Greedy Choice is to pick the smallest weight edge that does not cause a cycle in the MST constructed so far.

Kruskal's Algorithm



Prim's algorithm

Below are the steps for finding MST using Prim's algorithm -

1) Create a set *mstSet* that keeps track of vertices already included in MST.

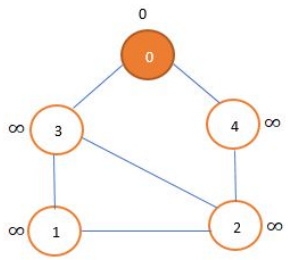
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.

3) While mstSet doesn't include all vertices

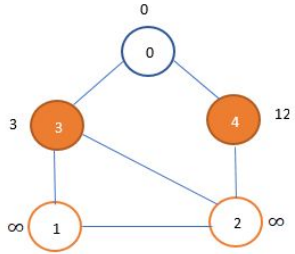
….**a)** Pick a vertex *u* which is not there in *mstSet* and has minimum key value.

….**b)** Include *u* to mstSet.

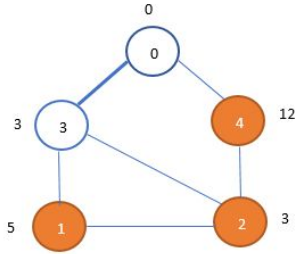
….**c)** Update key value of all adjacent vertices of *u*. To update the key values, iterate through all adjacent vertices. For every adjacent vertex *v*, if weight of edge *u*-*v* is less than the previous key value of *v*, update the key value as weight of *u*-*v*


0
0
∞
3
4
∞
∞
1
2
∞

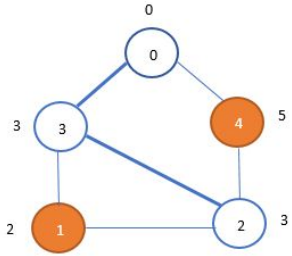
1


0
0
3
3
4
12
∞
1
2
∞

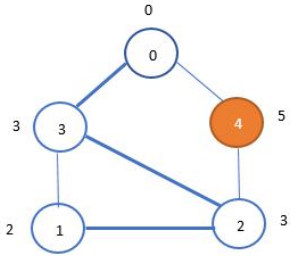
2


0
0
3
3
4
12
5
1
2
3

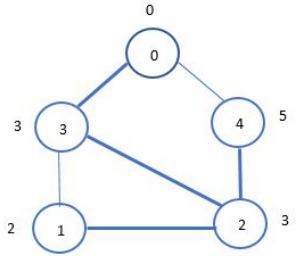
3


0
0
3
3
4
5
2
1
2
3

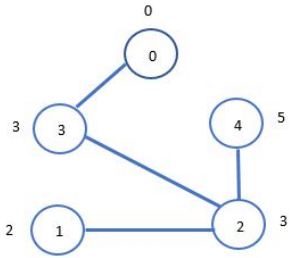
4


0
0
3
3
4
5
2
1
2
3

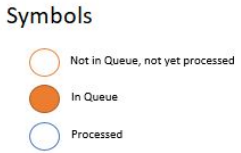
5


0
0
3
3
4
5
2
1
2
3

6


0
0
3
3
4
5
2
1
2
3

7


Symbols
Not in Queue, not yet processed
In Queue
Processed