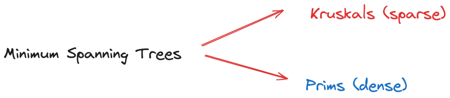
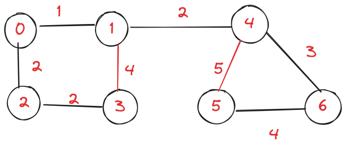


Minimum Spanning Trees

Kruskals (sparse)

Prims (dense)



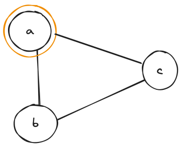
[[0, 1, 1] , [0,2,2], [2,3,2] , [1,3,4] , [1,4,2] .....]

Kruskals

Kruskals is done using the edge list representation of graph.

kruskals internally use DSU.

A tree is a graph without cycles. So to convert a graph into a tree, we need cycle detection. DSU can help u find cycles very easily

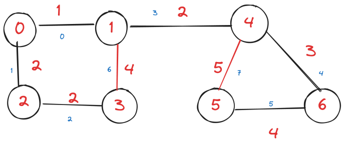


union(a, b)
union (b, c)4

union(a, c)

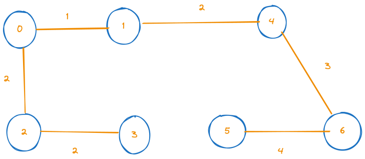
DSU can be used to do cycle detection, because DSU can check if two elements are in same group or not, if yes and we add a direct edge between then its a cycle.

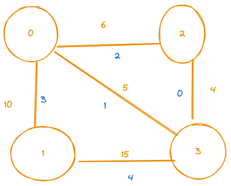
Sort the edge list based on weight (inc order)
- one by one keep picking the edges from smaller weight to larger weight
- while picking an edge check if it creates cycle or not using DSU
- And then if it doesn't creates a cycle add that edge to the ans.



Trees which are connected graphs always have N-1 edges

0	1	2	3	4	5	6
0	0	0	0	0	0	0



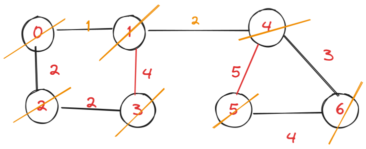


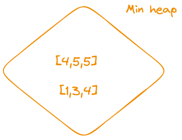
N = 4
3 edges

4+5+10

Prims

BFS + PQ





Min heap

[0,1, 1]

[0,2, 2]

[2,3,2]

[1.4.2]

[4,6,3]

[6,5,4]