Date   Page

Day - 215

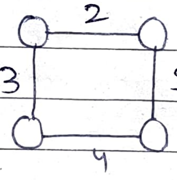
Graph - 19

* **Spanning Tree:**

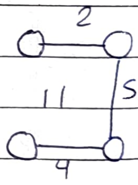
⇒ It is a subset of Edges of the graph that forms a tree where every node of the graph is a part of tree.

⇒ So, Tree: 'n' node, then 'n-1' edges
⇒ No Cycle
⇒ All nodes are connected

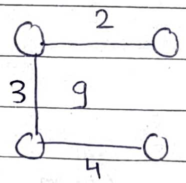
Graph:



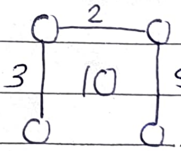
→ Spanning Tree:



⇒ Also,



,

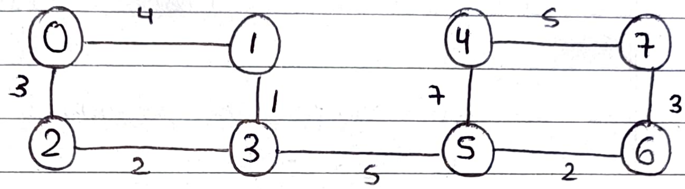


⇒ That means, a tree can have many spanning tr~~e~~ees.

⇒ So, minimum spanning tree is that have min. weight out of every spanning tree.

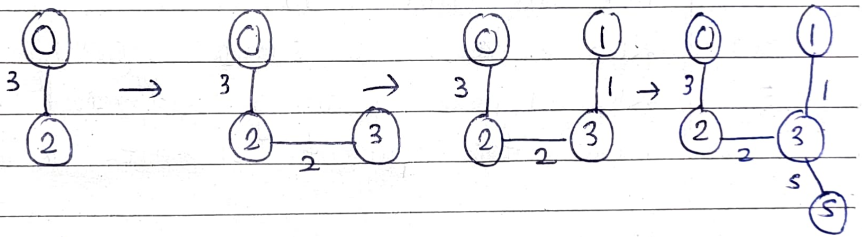
⇒ In the real world if we want min. length of something to connect things like railway lines, cable lines, etc.

* Prim's Algo:

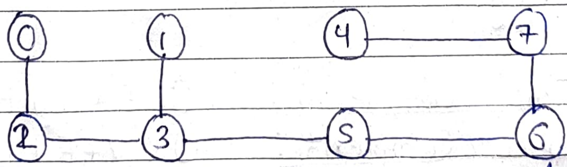


=> We will start from source and select the min. path out of all available weighted edges & path.

=>



=> and so on —



=> Also, don't form any cycle.

=> If you get any cycle, reject that path.

=> we require a mst array that ~~gives us~~ is used to check if that node is part of mst or not.

=> We also require min-heap or priority queue.

to select the min edge.

⇒ Information will store in min-heap i.e. (wt, node, parent).

⇒ we also make a parent array.

⇒ So, we will start by pushing source node i.e. (0, 0, -1).

⇒ Then pop it, and check for part of the mst or not then if not, make mst i.e. 1 of source.

⇒ After that put their adjacent nodes into the min heap.

⇒ And follow the same process.

⇒ Also, before pushing adjacent nodes, make the parent of that node.

⇒ Parent array helps in creating the mst easily.

Time Complexity: O(E * log E)
Space Complexity: O(V+E)