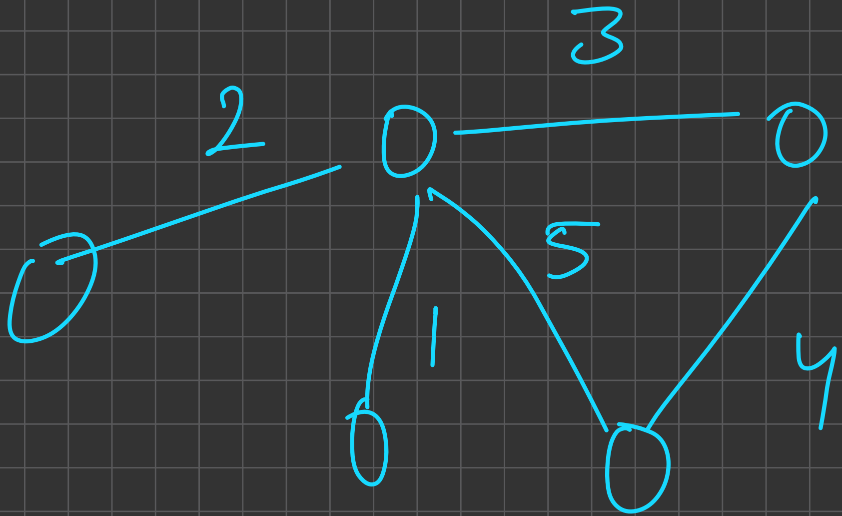
Graph Theory

Algorithms :

① Kruskal's Algo

② Prim's Alg

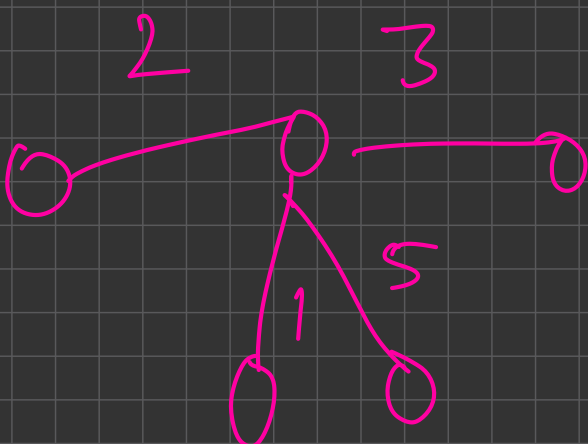
Minimum Spanning Trees Algorithms and Problem Solving



spanning tree

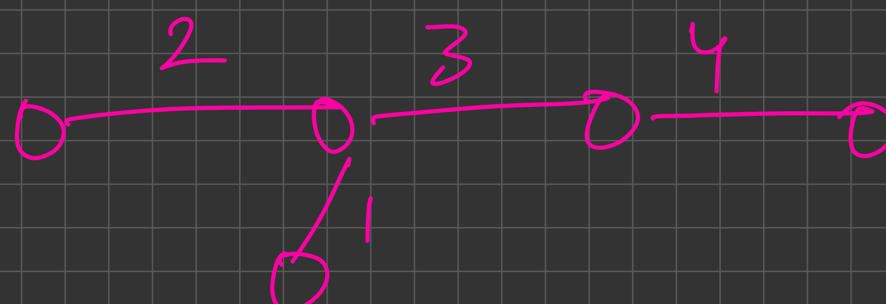
n nodes m edges

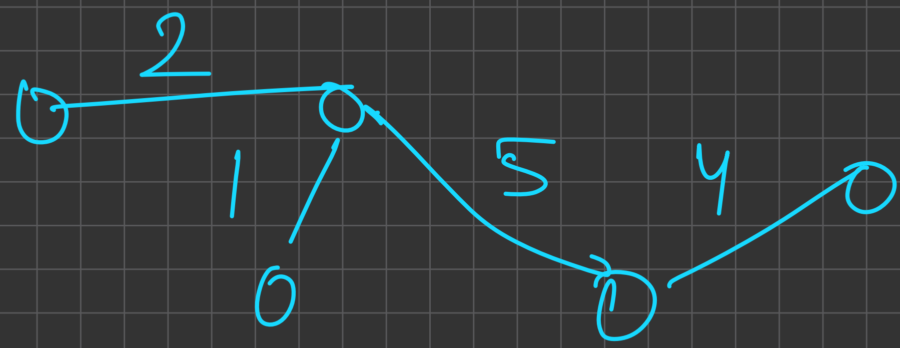
minimum S.T



$m >> n$

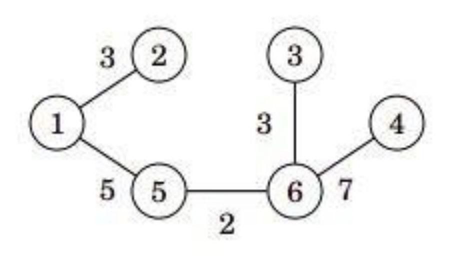
maximum S.T




2
1
0
5
4

Minimum Spanning Tree

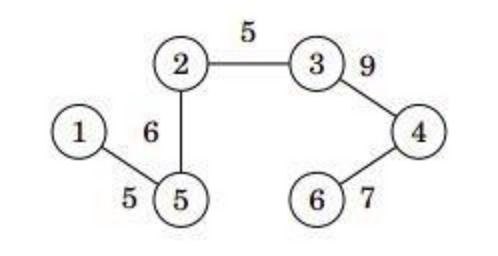
A minimum spanning tree is a spanning tree whose weight is as small as possible.



The weight of the minimum spanning tree for the example graph is 20.

Maximum Spanning Tree

A maximum spanning tree is a spanning tree whose weight is as large as possible.



The weight of a maximum spanning tree for the example graph is 32.

Minimum Spanning Tree Algorithm

(1) Kruskal's Algorithm

(1) Sort all the edges by their weight

$m \log m$ (m = # edges)

(2) Iterate over all the edges and consider the edge if it doesn't make a cycle

— verifying an edge $O(n+m)$

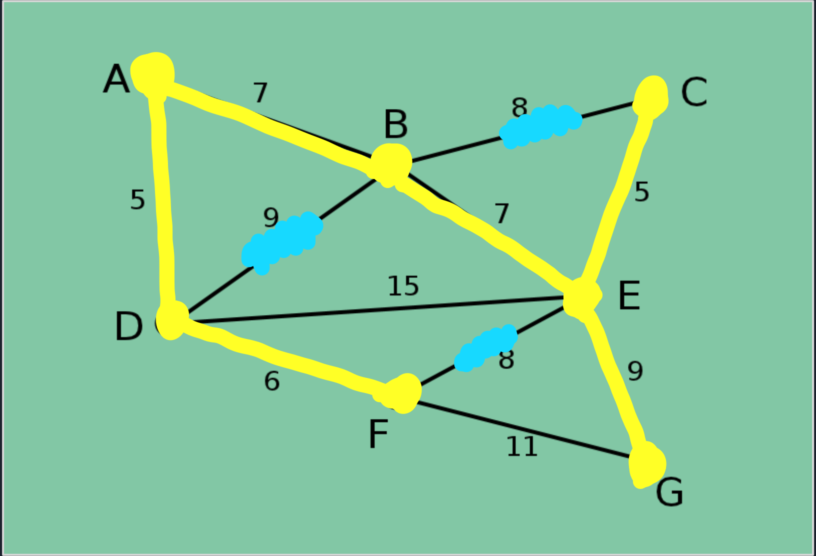
Kruskal’s Algorithm

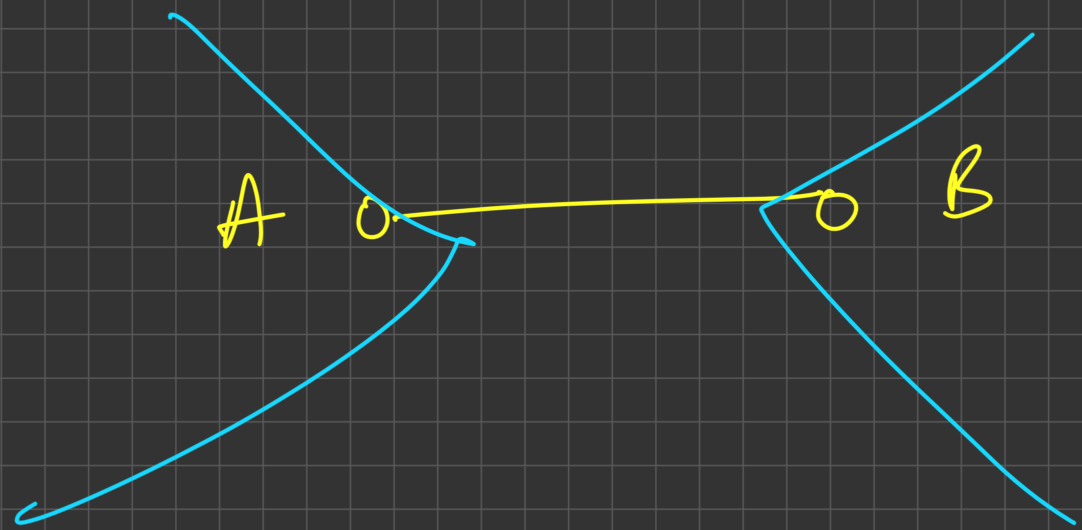
Kruskal’s algorithm is a greedy algorithm to find the minimum spanning tree of a graph. The algorithm works as follows:

1. Sort all the edges from low weight to high.
2. Take the edge with the lowest weight and add it to the spanning tree. If adding the edge created a cycle, then reject this edge.
3. Keep adding edges until we reach all vertices.

Kruskal’s Algorithm

Consider the given graph:




A
O
O
B

① Sort the edge list edge {a, b, c}

for (edge : edges) {

⟹ if a and b are in same component
continue $O(1)$

} merge (a, b) , weight += c $O(1)$

$O(\log^* n)$

$N \le 10^5$

$\boxed{\log_2 10^5} = 20$

$\log_2 10^5 \longrightarrow \log_2 20 \longrightarrow \log_2 5 \longrightarrow \log_2 2$

$\longrightarrow \log_2 1 \longrightarrow 0$

Assume the M.S.T of a graph is

$\underline{\underline{T}}$

Global known M.S.T $\longrightarrow$ T

Your Kruskal's M.S.T $\longrightarrow$ $\underline{\underline{K}}$

1 2 3

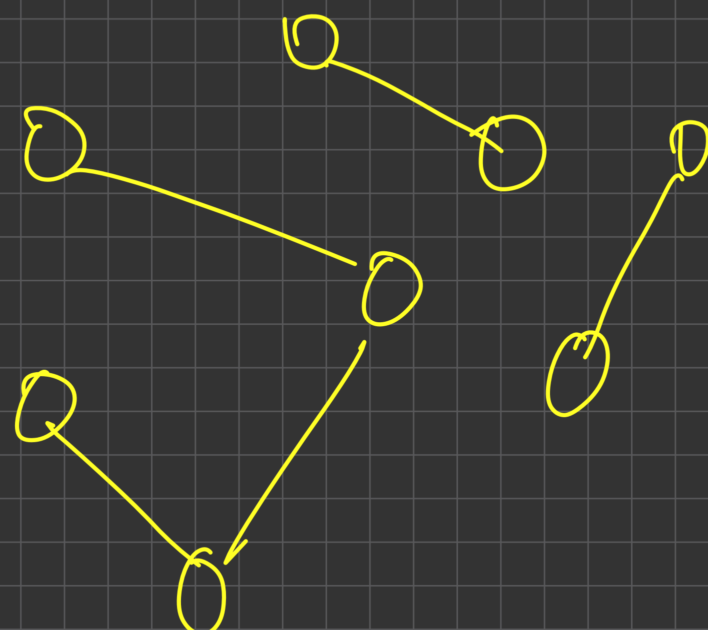
At every step of Kruskal's Algo our current set of edges will be a subset of the global MST?

0 th step

$\{ \quad \}$

Let's assume that after x steps all the edges added so far are the subset of the global M.S.T

$x = 5$



$\underline{\underline{x = 6}}$

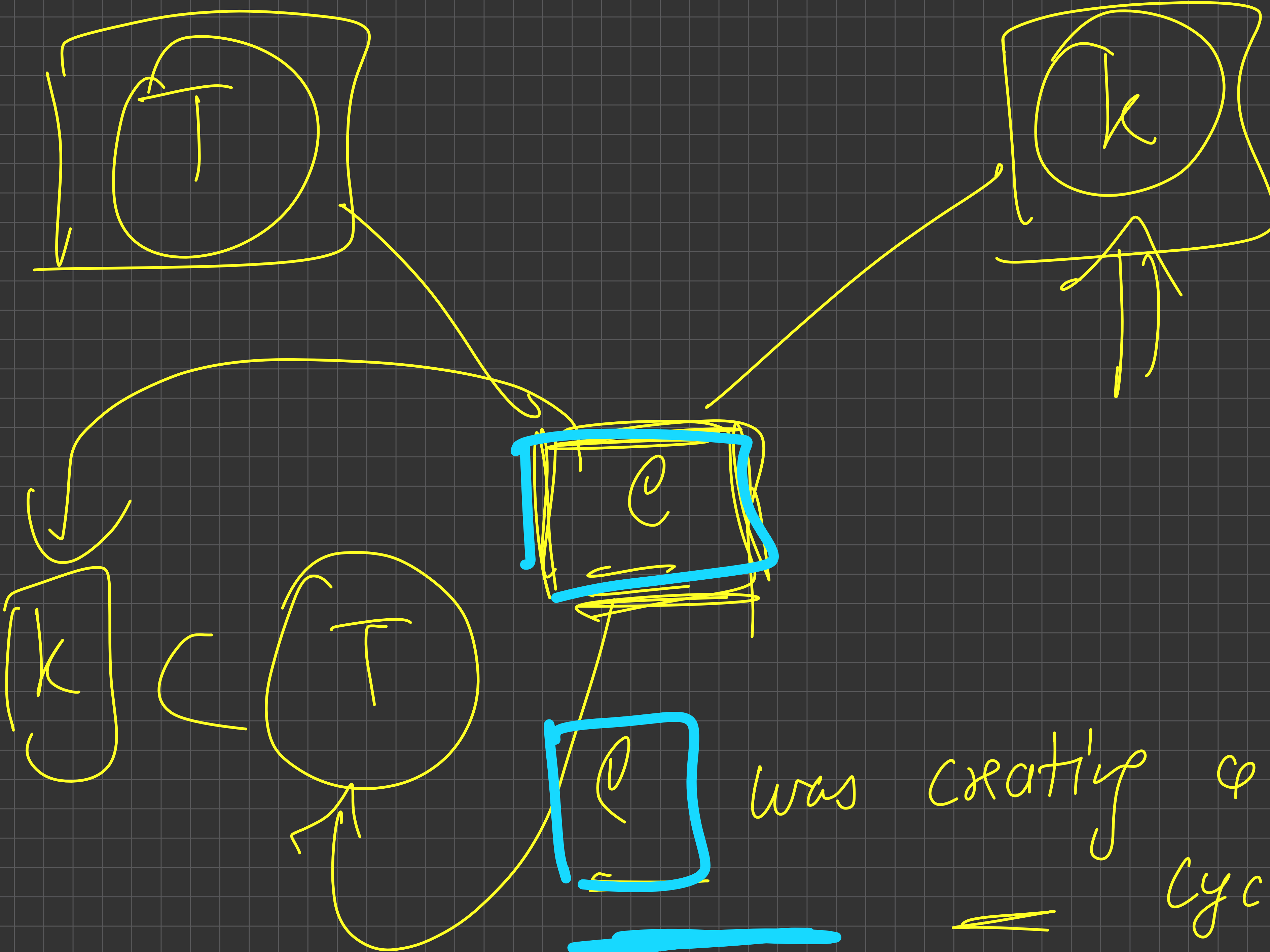
After x steps all the x edges are in the global M.S.T.

T

$\boxed{K}$

x edges

$= \Sigma$

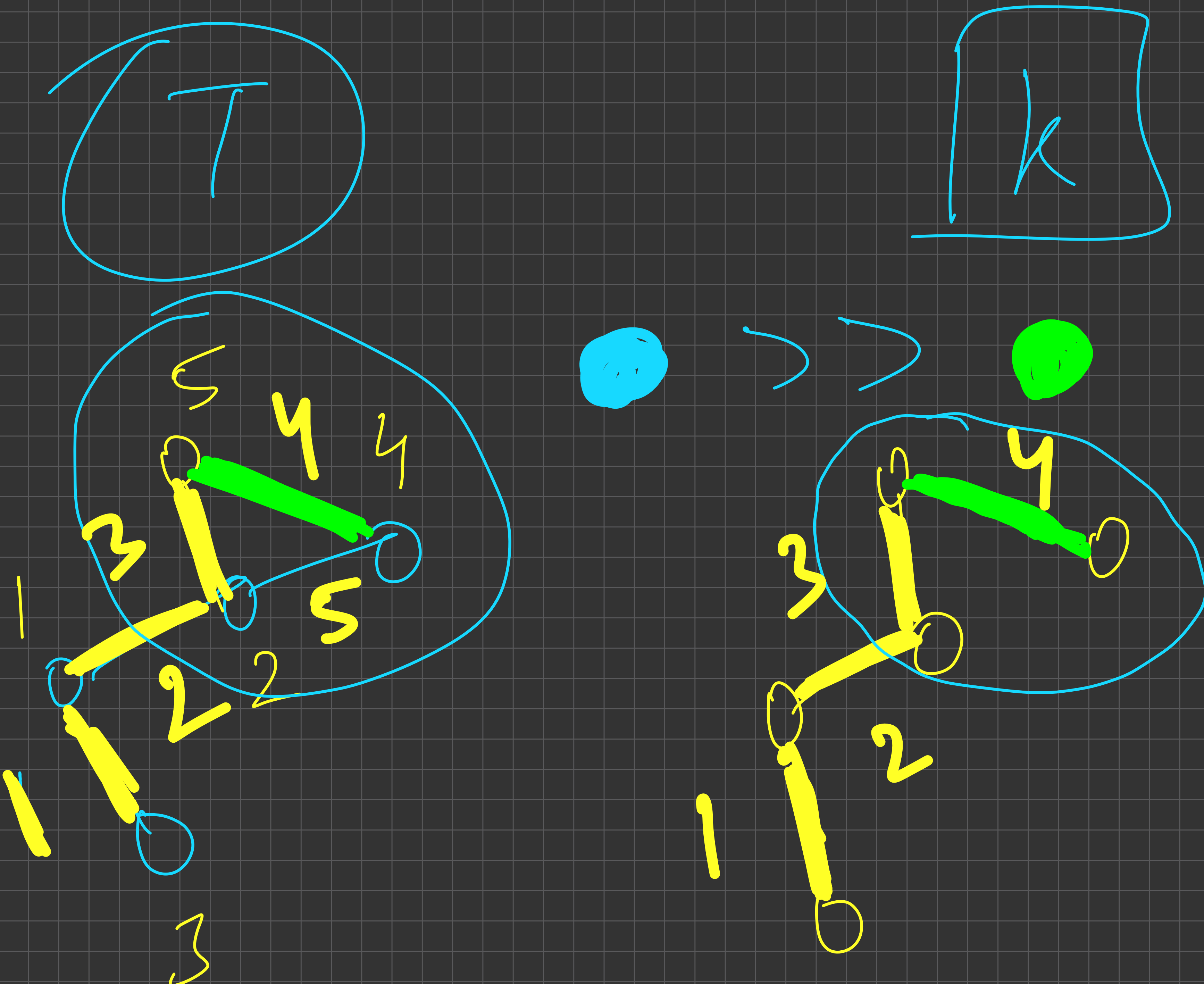


was creating a cycle in k

$e \rightarrow$ present in both K & T ✓

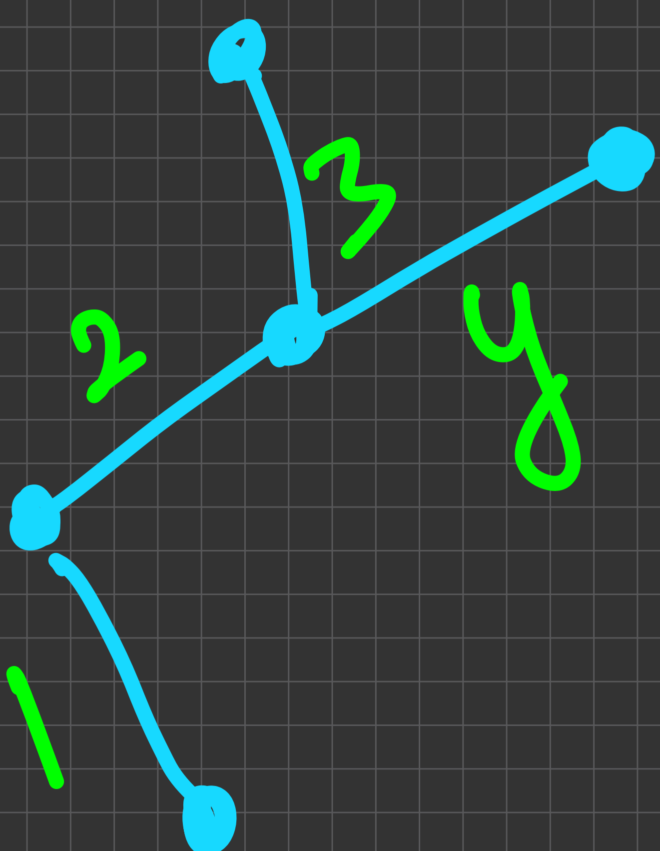
$e \rightarrow$ creating a cycle in K ✓

$e \rightarrow$ not creating a cycle in K
but it is not present in T

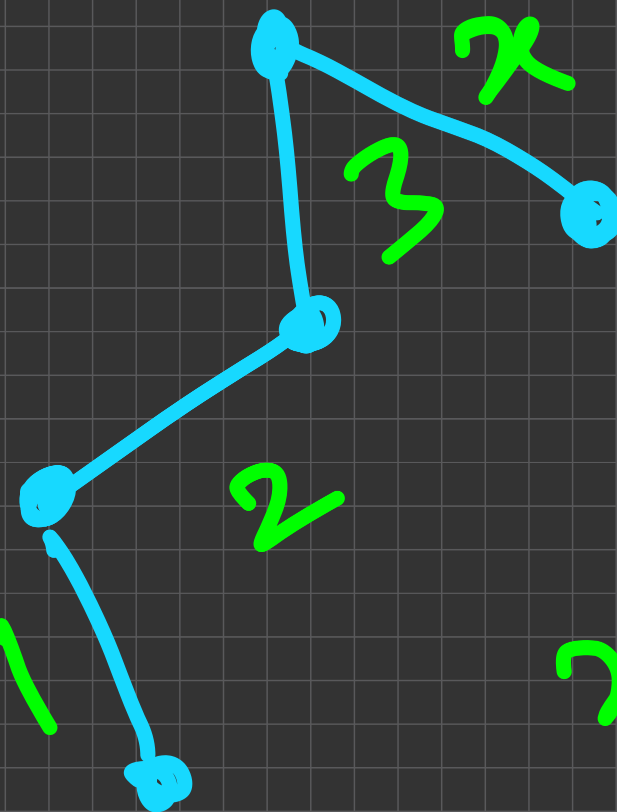


T
k
5
4
4
3
1
5
2
2
1
3
4
3
2
1

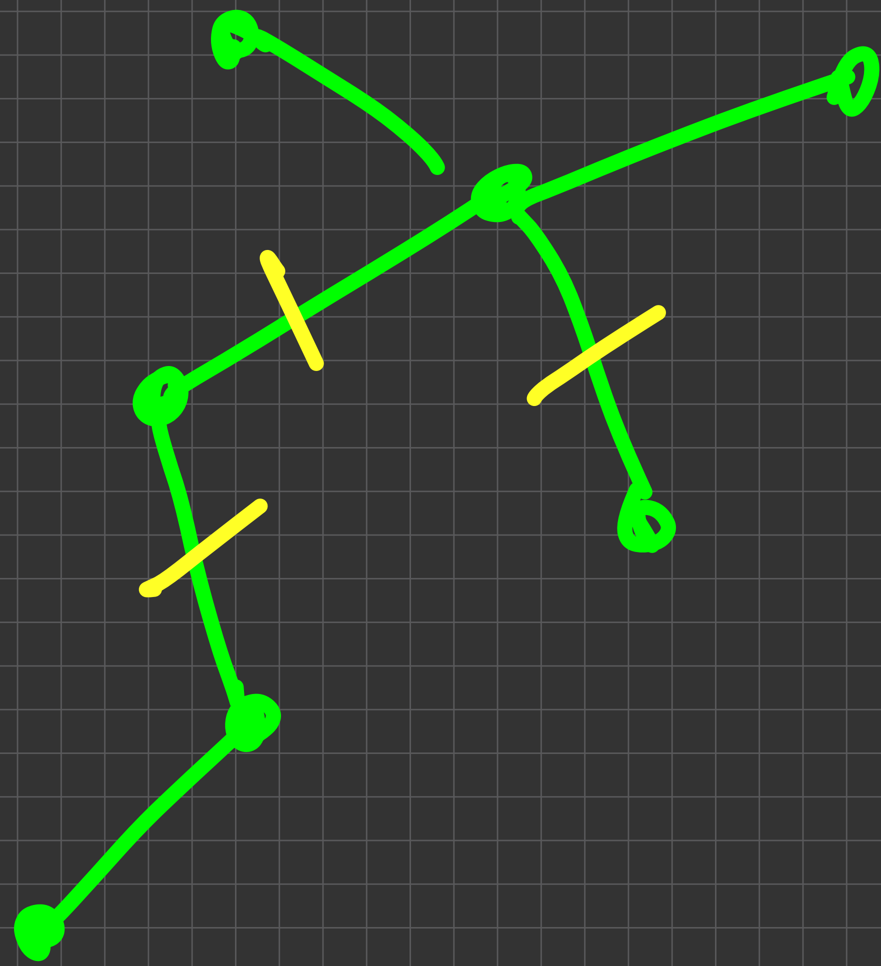
T

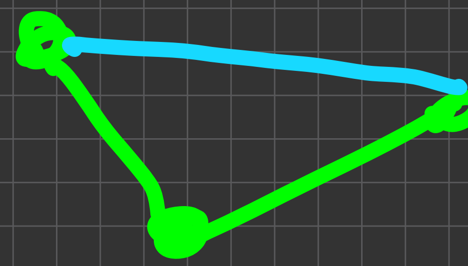


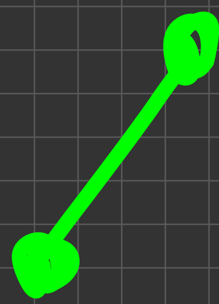
K

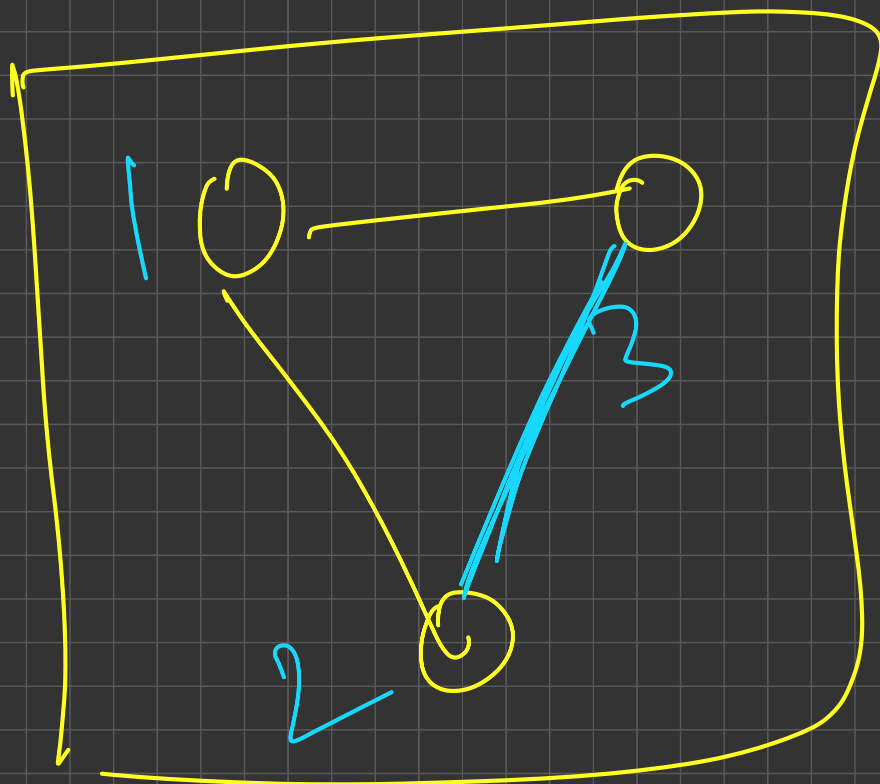


$x < y$

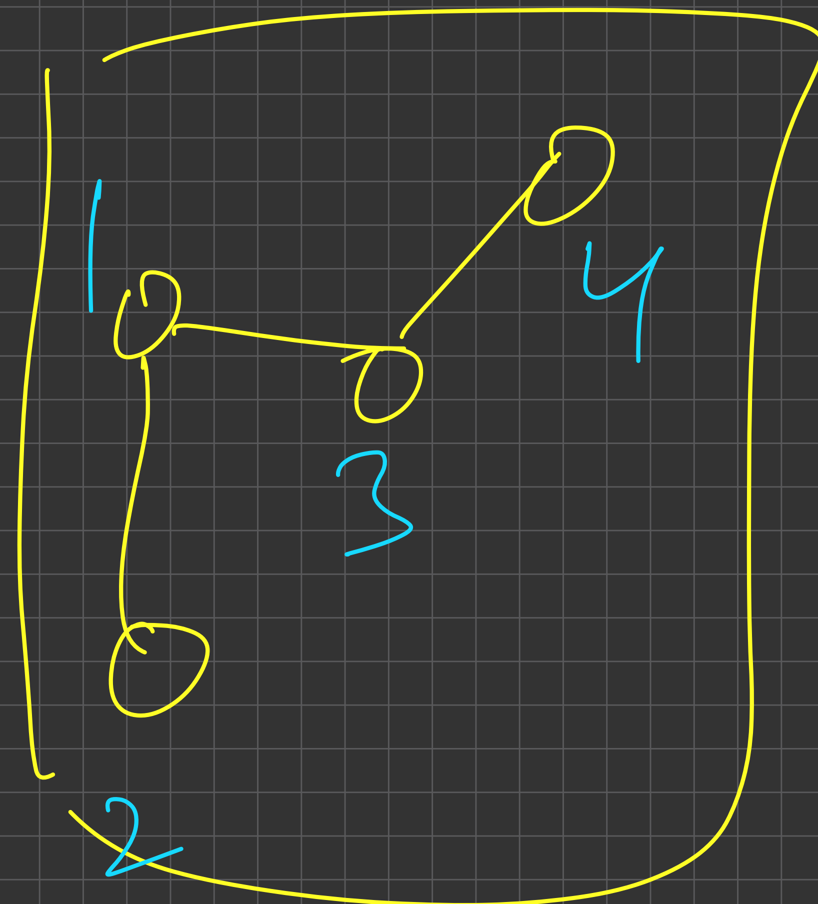








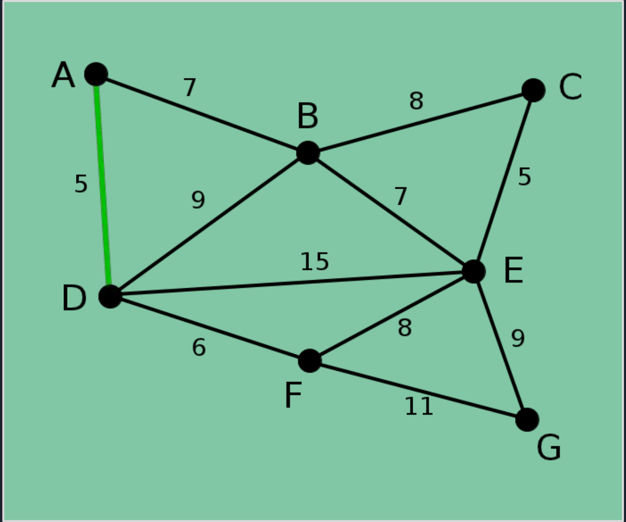
K

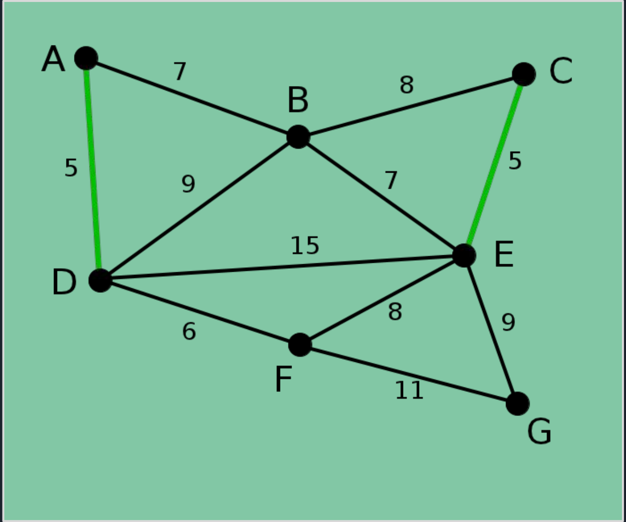


T

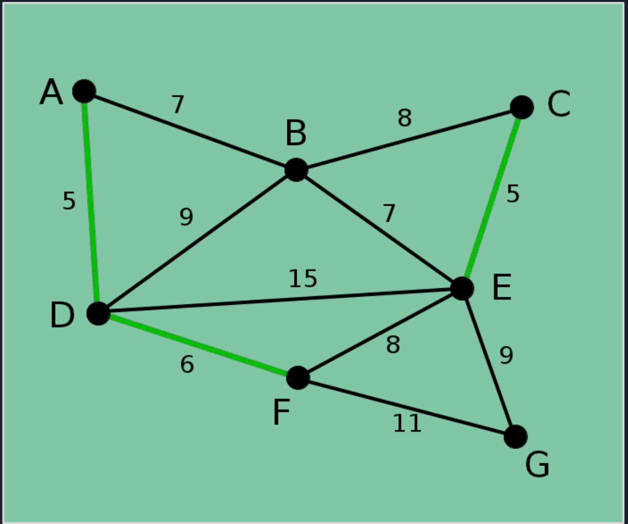
Smallest sec thir kth

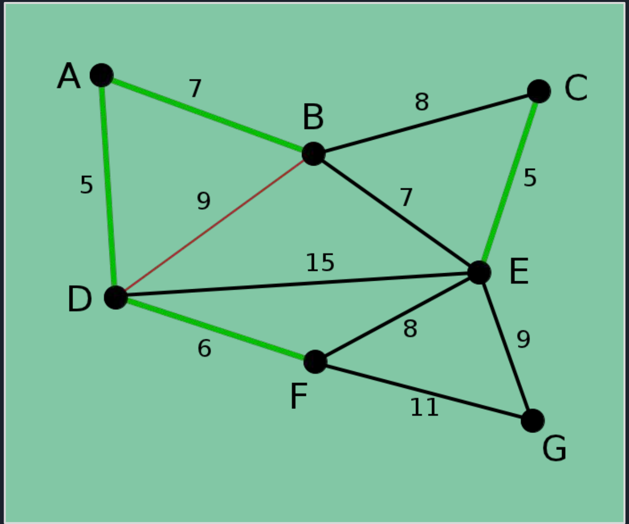
Kruskal’s Algorithm





Kruskal’s Algorithm



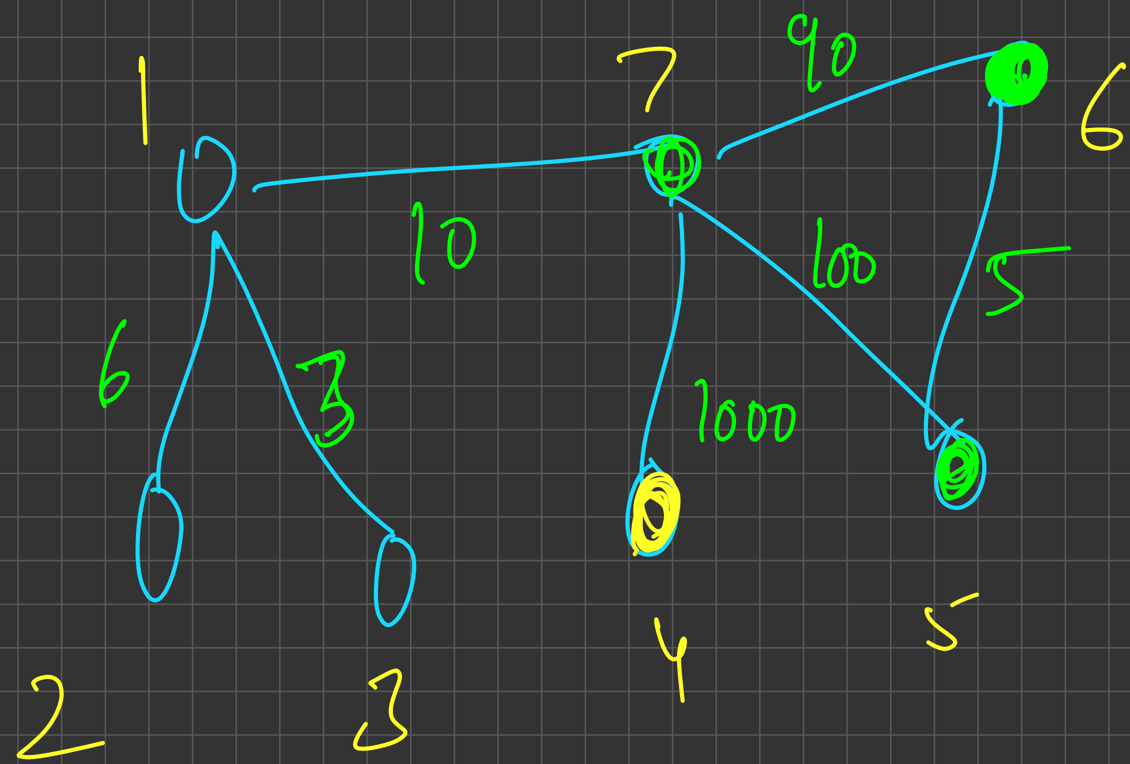


A city $\longrightarrow$ n houses $(\leq 10^9$

$n \leq 10^5$

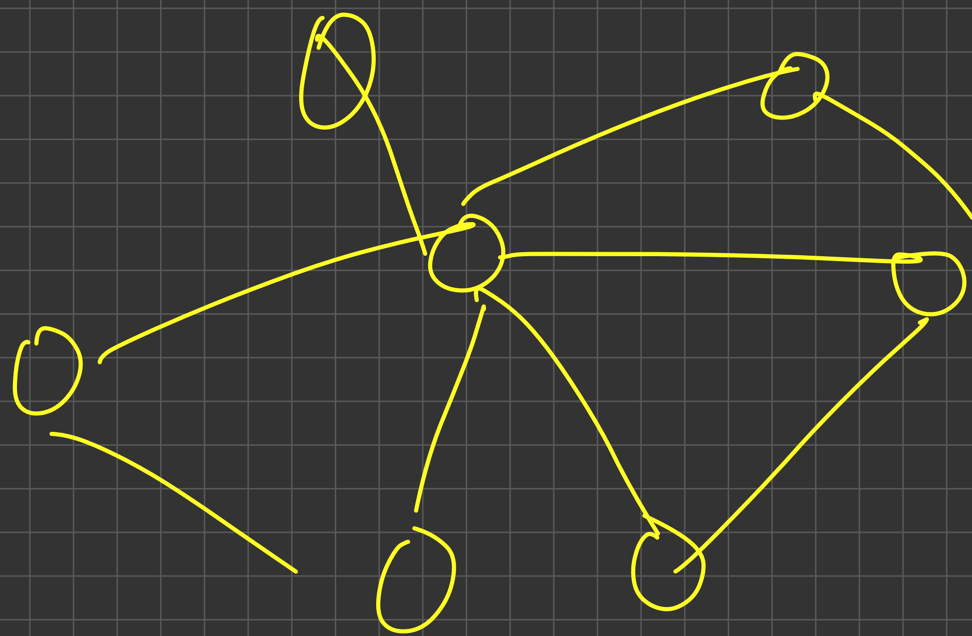
$m \leq 10^5$

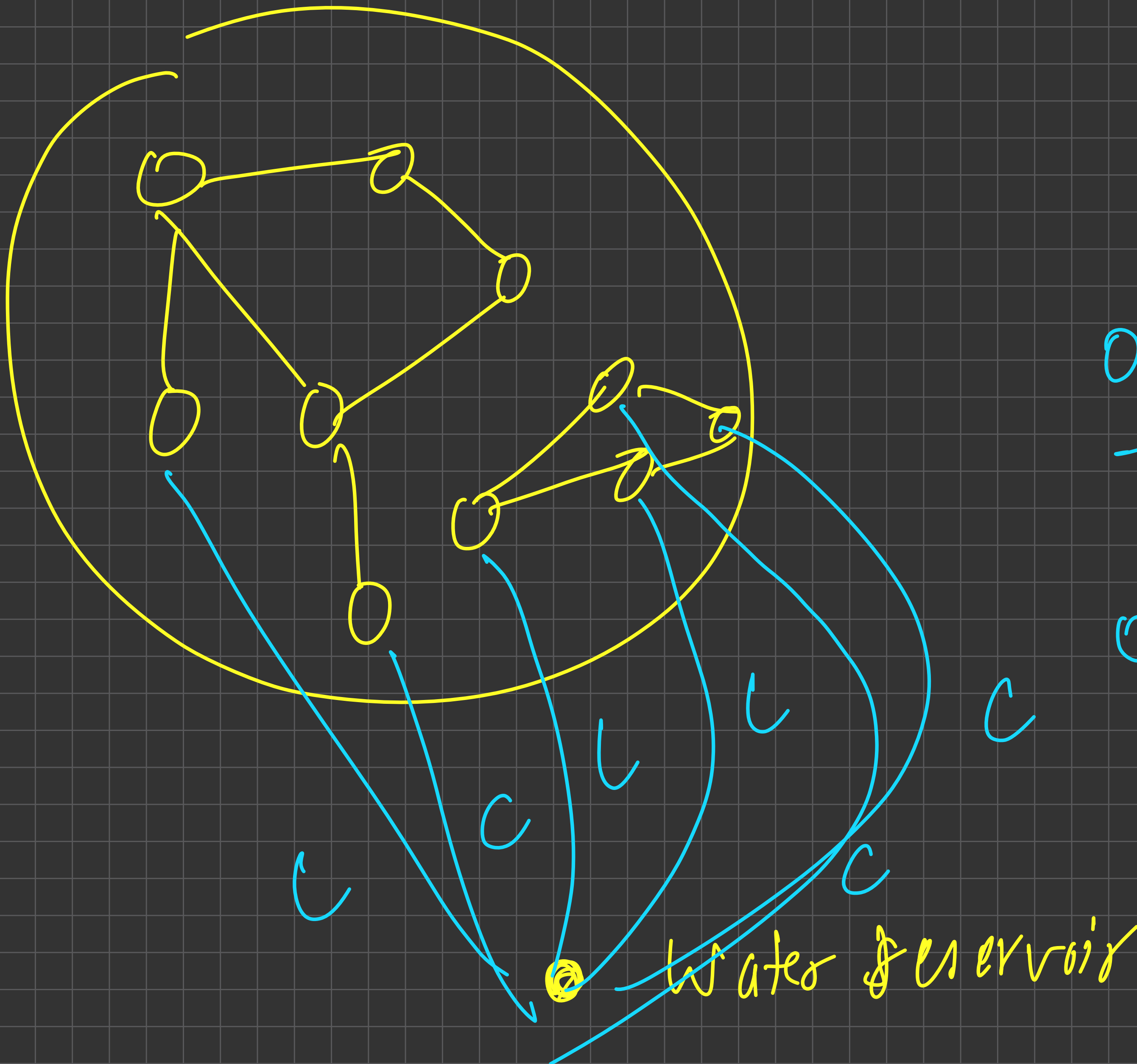
pipes b/w houses



100

+95

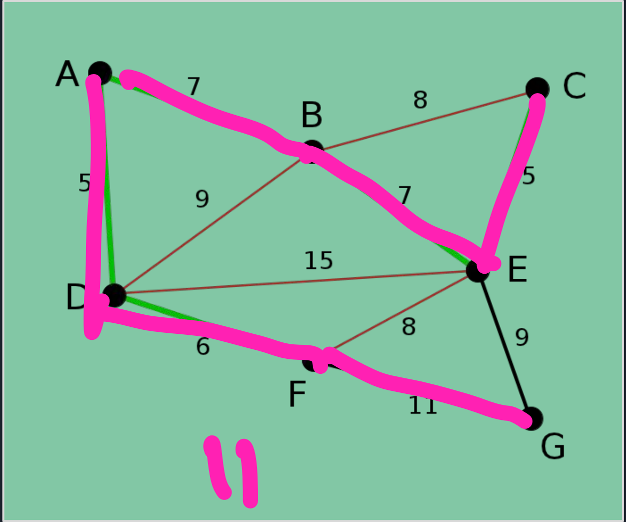


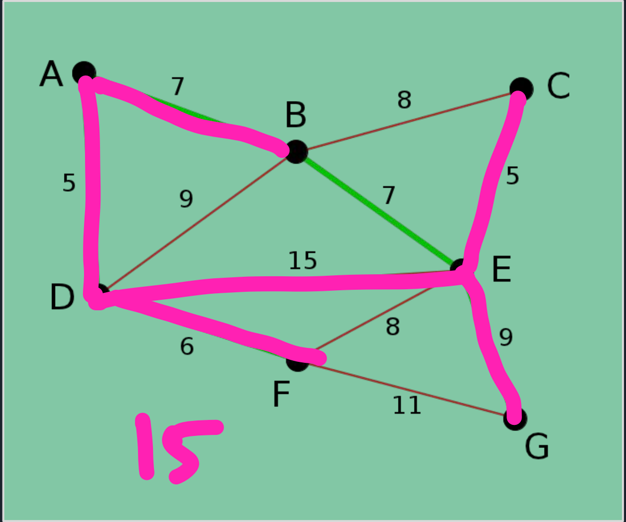


$O(n+m)$

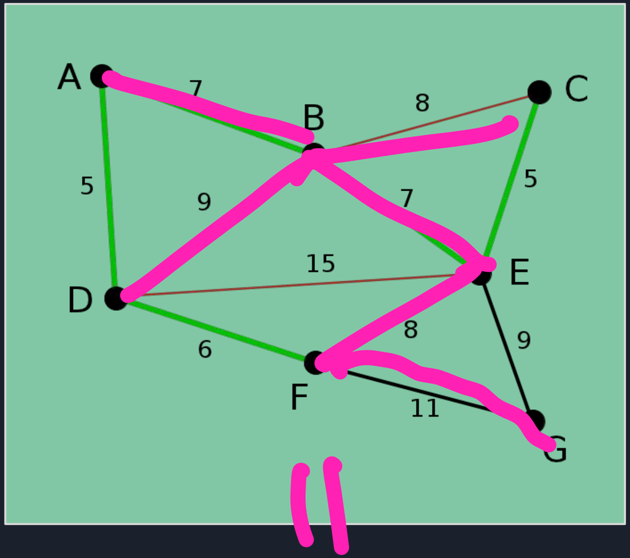
$O(n+m \cdot \log m)$

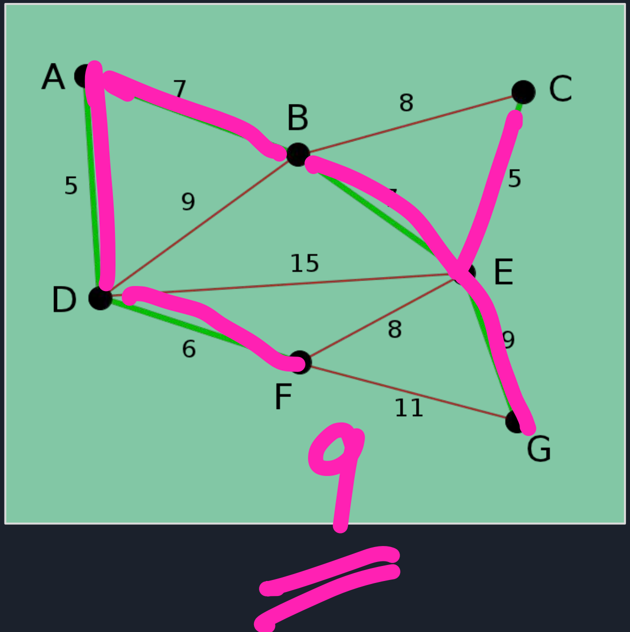
Kruskal’s Algorithm





Kruskal's Algorithm





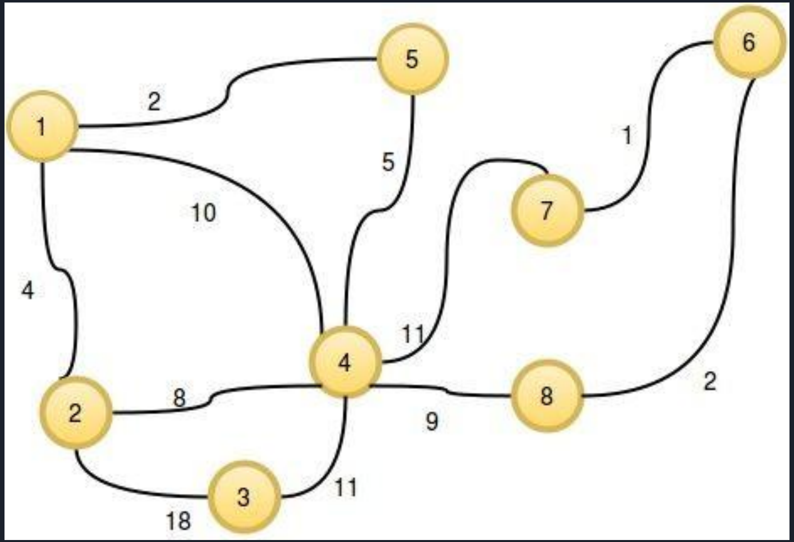
Prim’s Algorithm

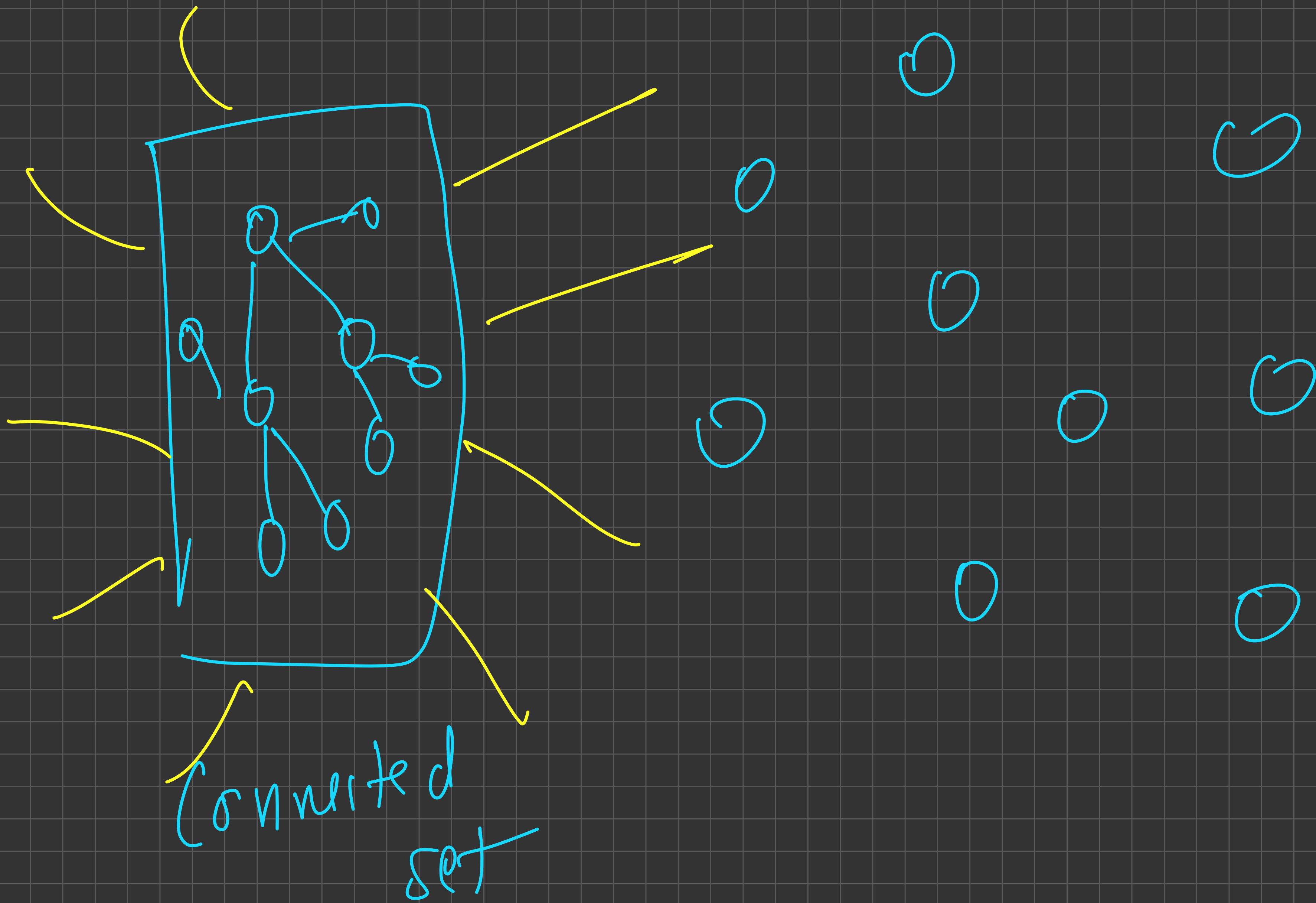
Prim’s algorithm is also a greedy algorithm to find the minimum spanning tree of a graph. The algorithm works as follows:

1. Initialize the minimum spanning tree with a vertex chosen at random.
2. Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree.
3. Keep repeating step 2 until we get a minimum spanning tree.

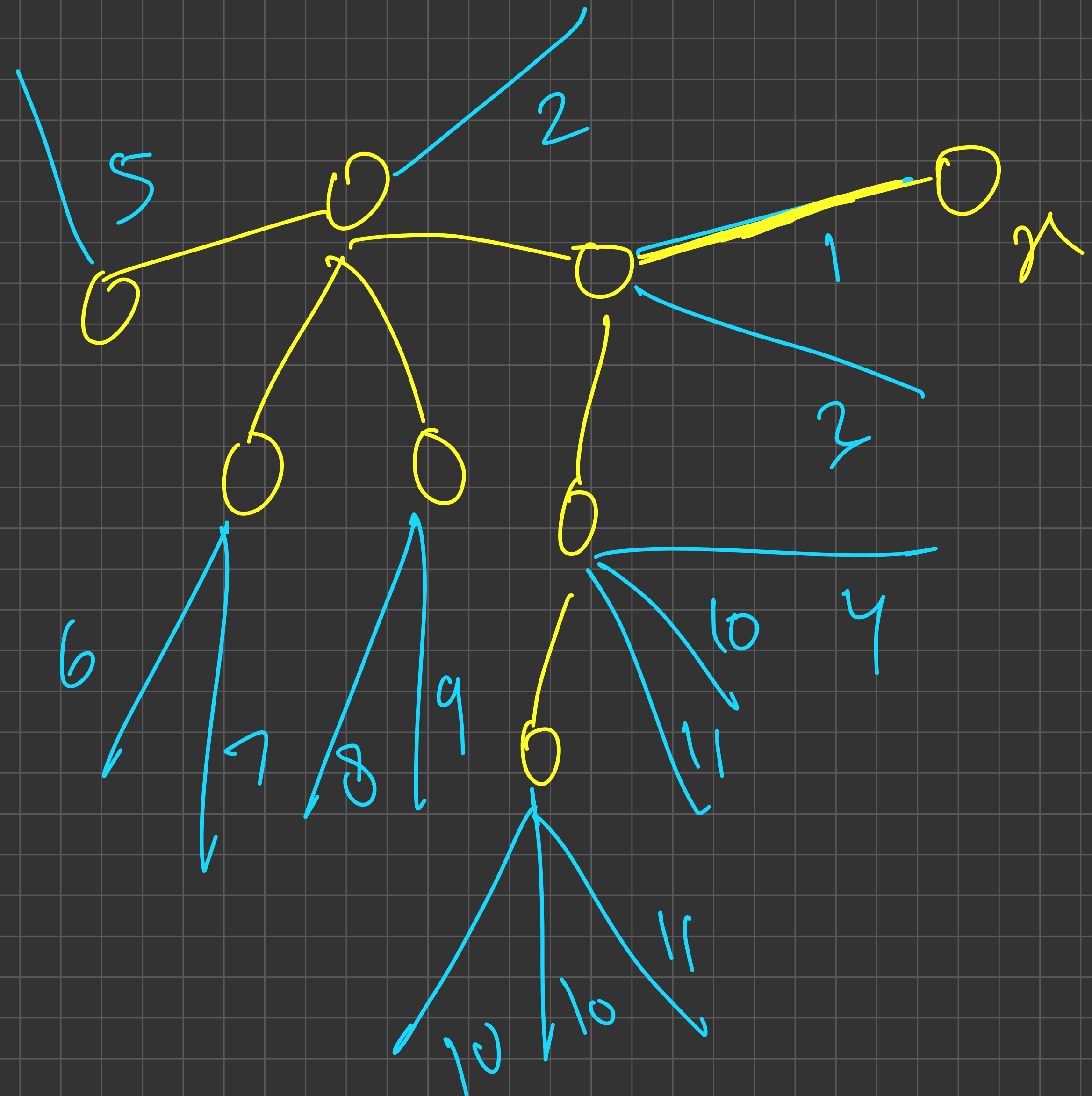
Prim’s Algorithm

Consider the given graph:





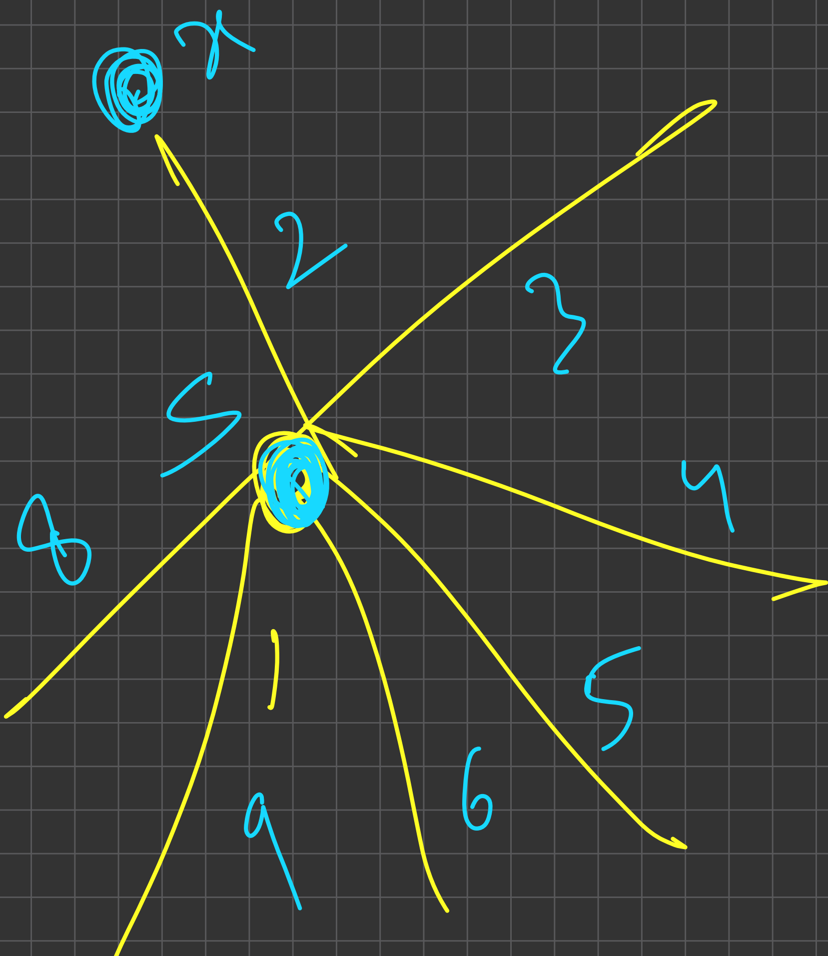
Connected
set

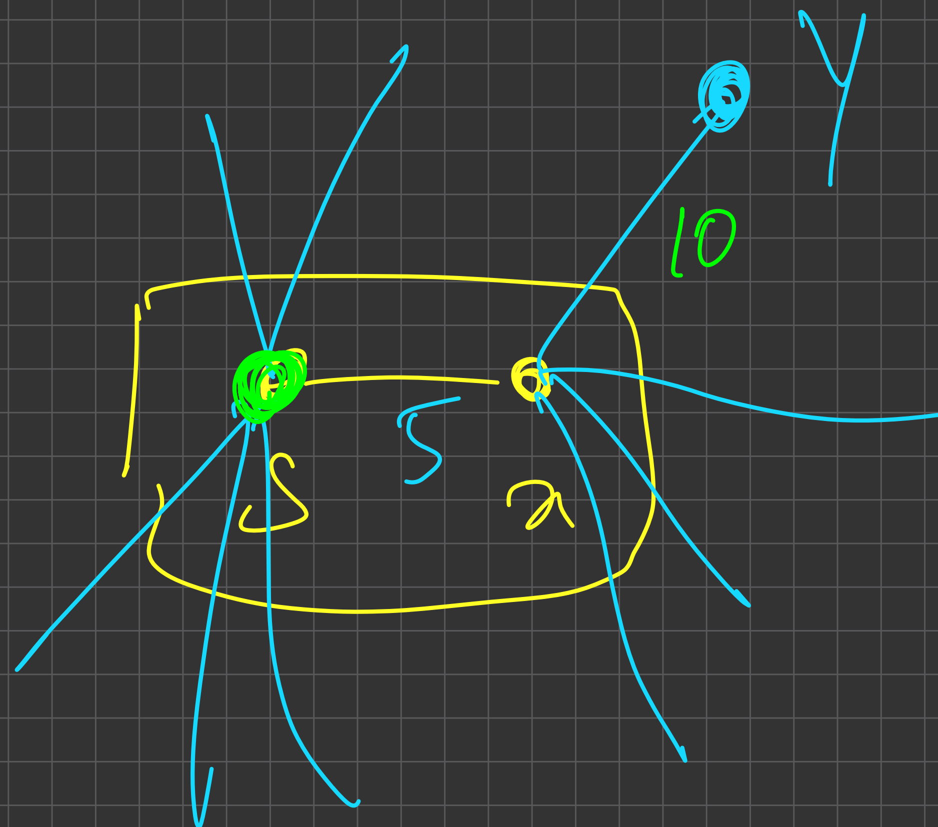


Prim's Algo

doesn't work for negative edge weights

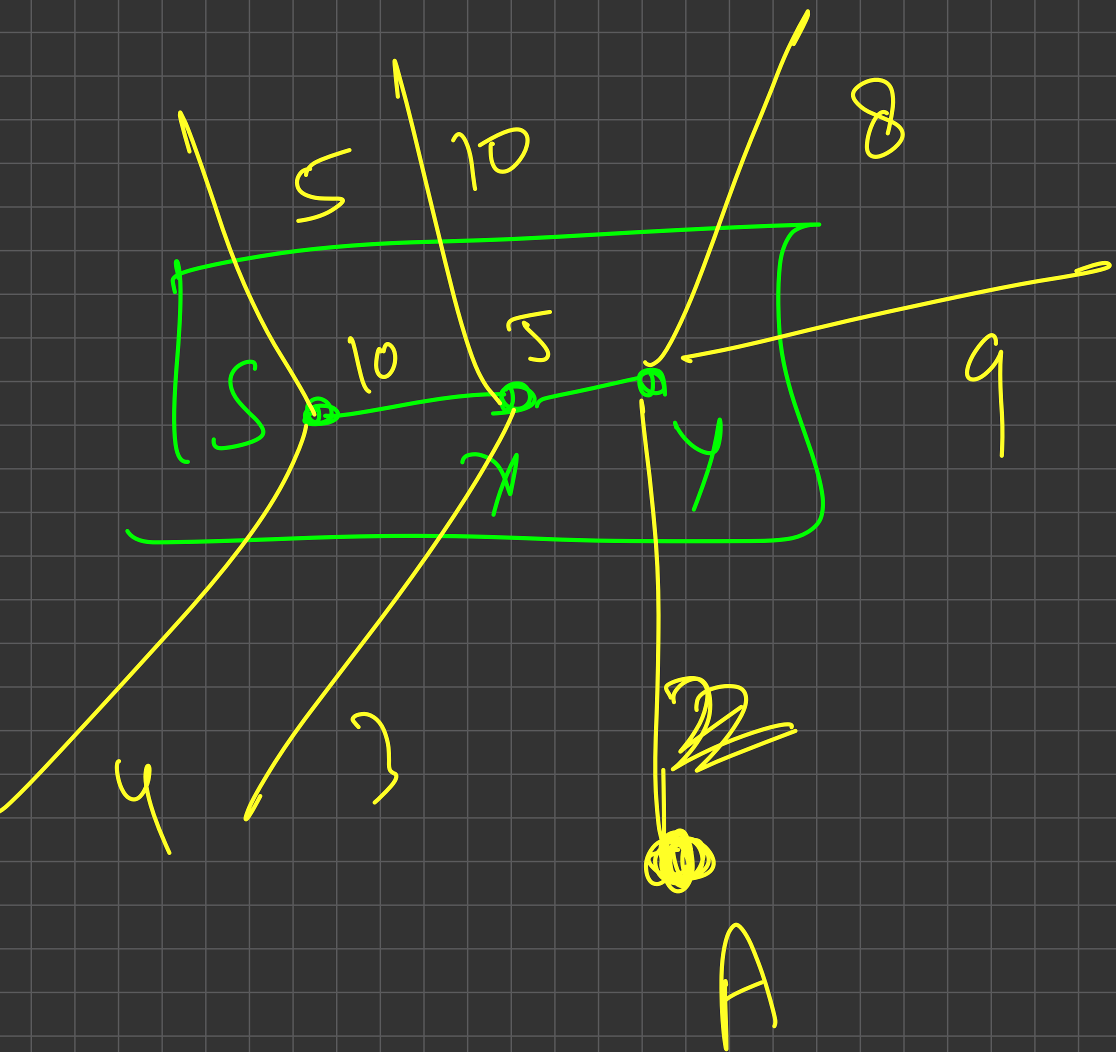
Start with random node



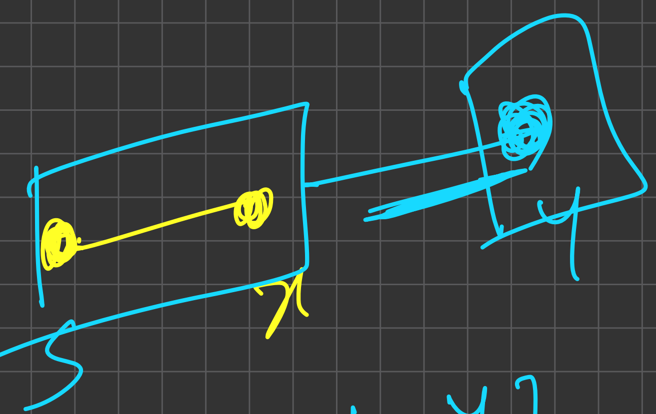


Dijkstra → S to Y

MST → connecting Y to MST


5
10
8
S
10
5
9
7
4
4
3
22
A

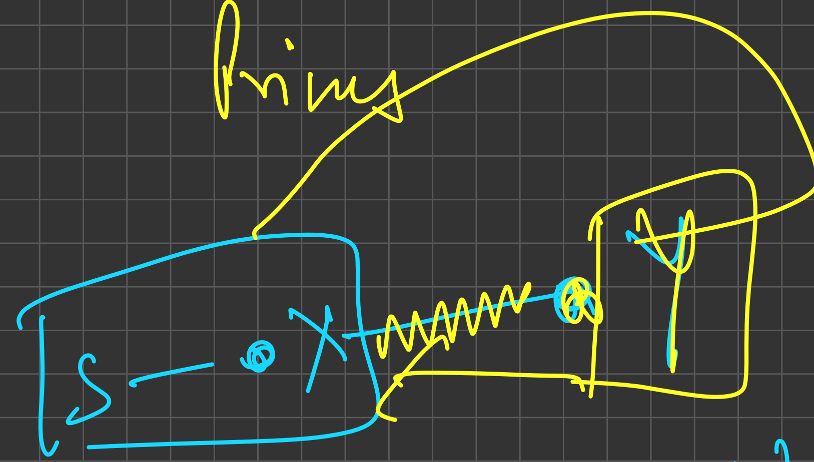
Dijkstra



edge [s to y]

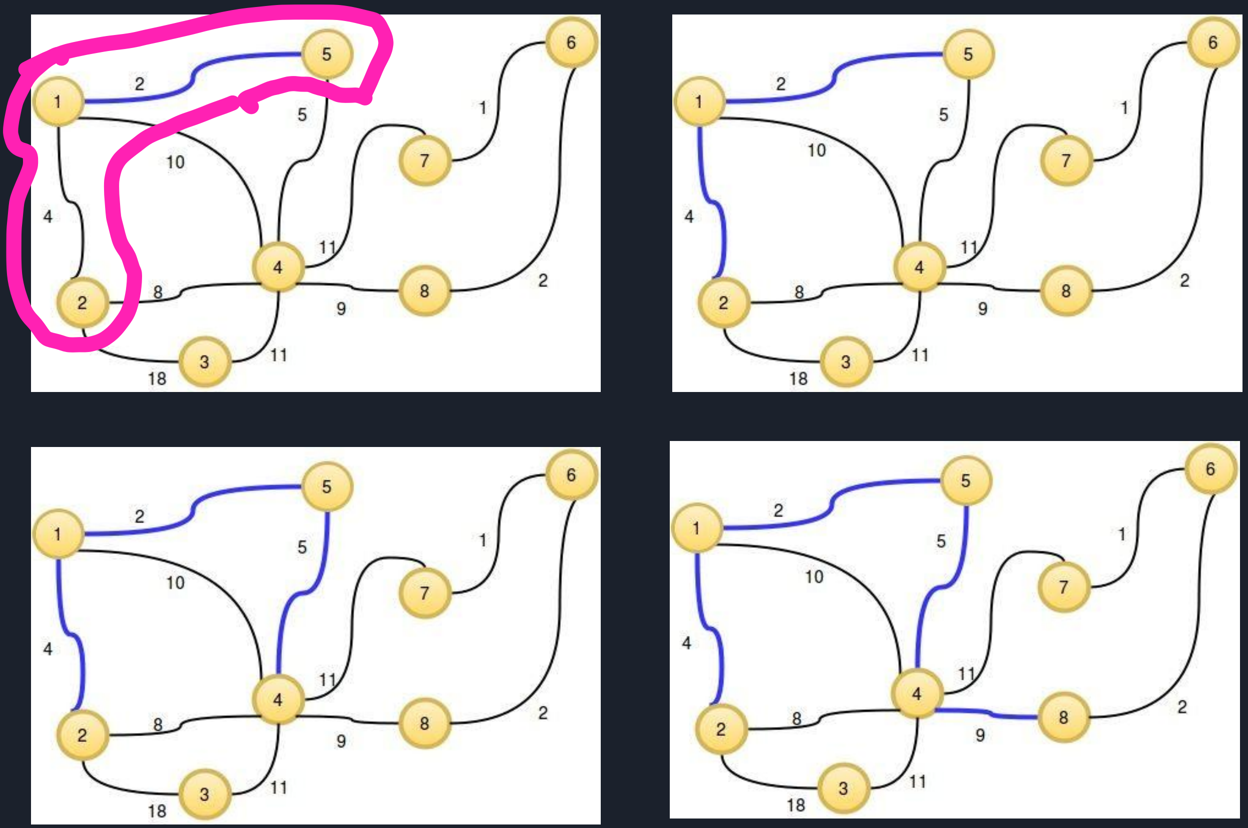
edge [x to y] + dist [x]

Prims



(edge [x to y])

Prim’s Algorithm



$\boxed{\text{Dijkstra}}$

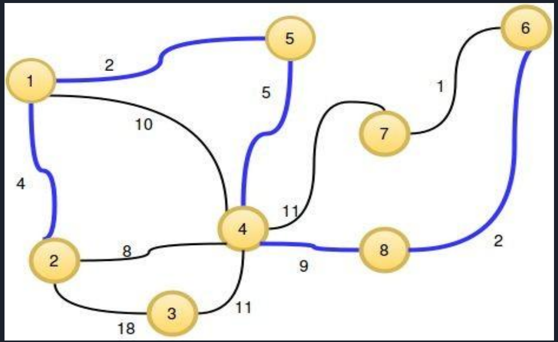
Kruskal's

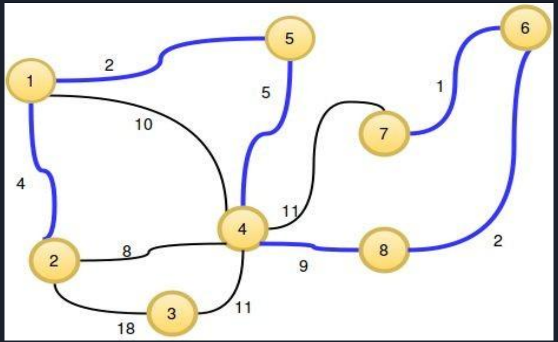
$\boxed{\text{Prim's}}$

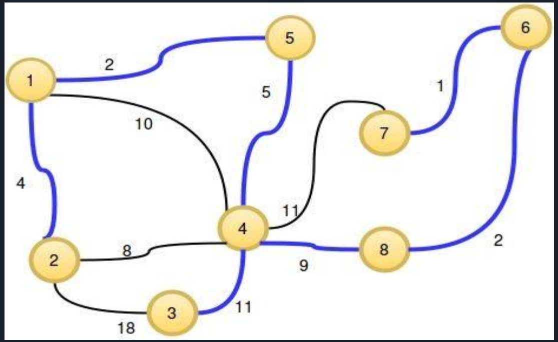
Multi Src

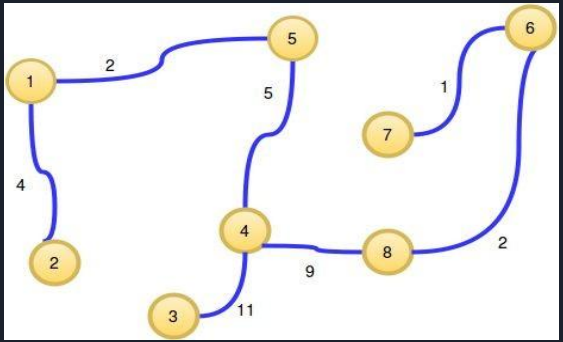
DSU

Prim’s Algorithm









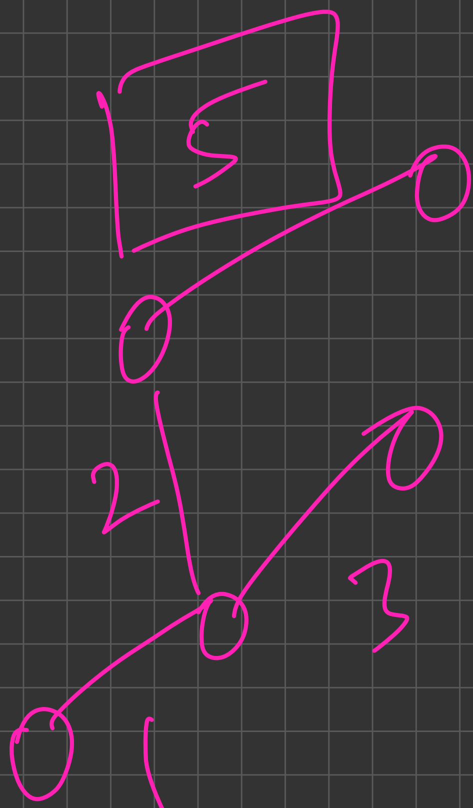
Some Properties of MSTs

Run Kruskal's Algorithm

- A minimum spanning tree of a graph is unique, if the weight of all the edges are distinct.
- Minimum spanning tree is also the tree with minimum product of weights of edges.
- In a minimum spanning tree of a graph, the maximum weight of an edge is the minimum possible from all possible spanning trees of that graph.

Some Properties of MSTs

- A minimum spanning tree of a graph is unique, if the weight of all the edges are distinct.
- Minimum spanning tree is also the tree with minimum product of weights of edges.
- In a minimum spanning tree of a graph, the maximum weight of an edge is the minimum possible from all possible spanning trees of that graph.



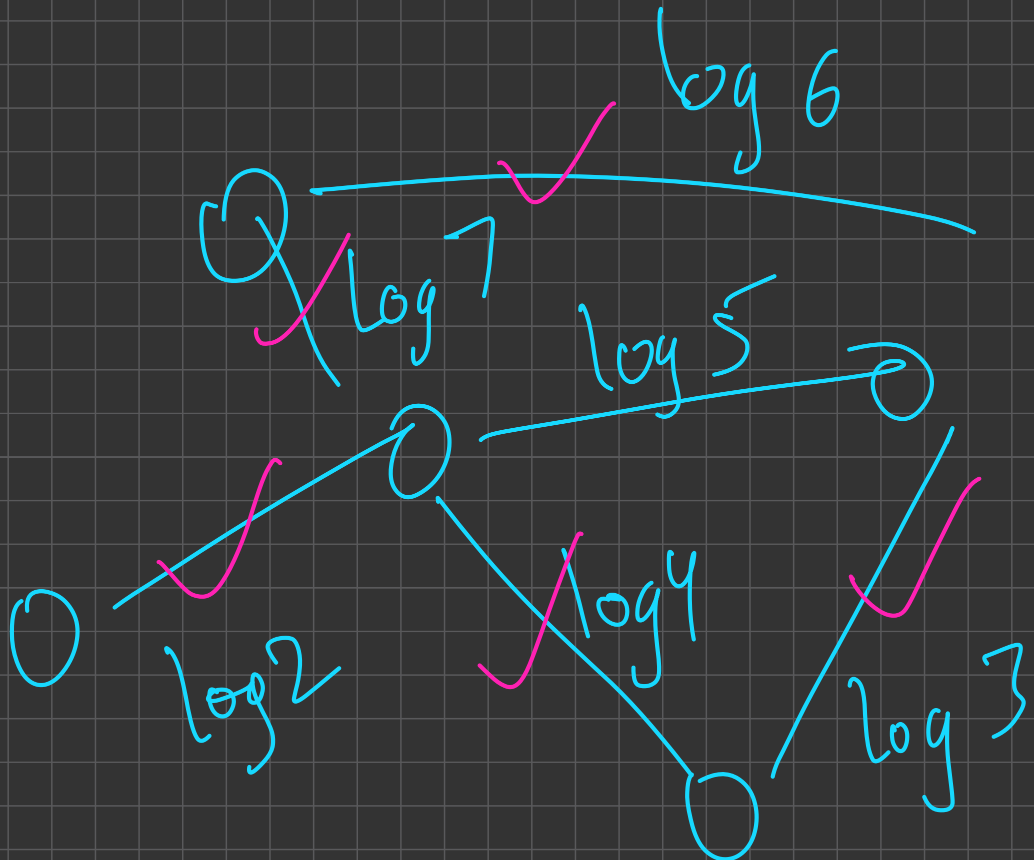
5

=

0

M.S.T → $\boxed{\text{~~Minimum Sum~~}}$

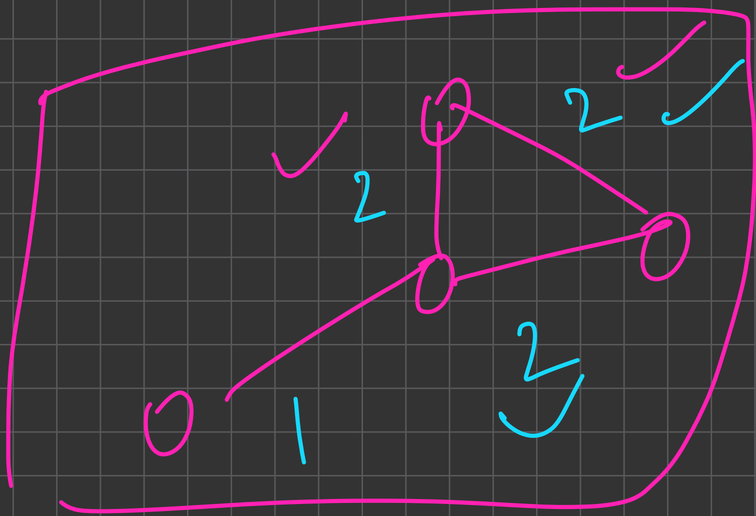
Minimum Product

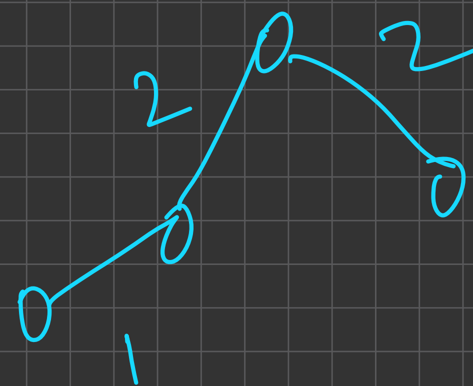


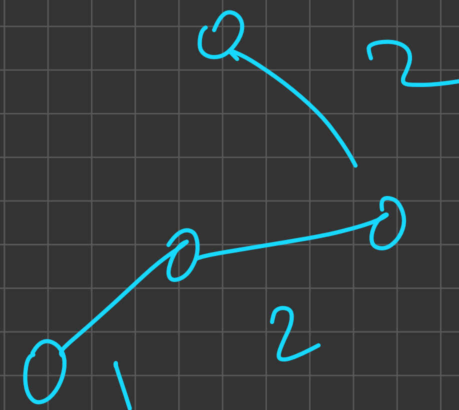
$$\log(2 \times 7 \times 3 \times 4 \times 6$$

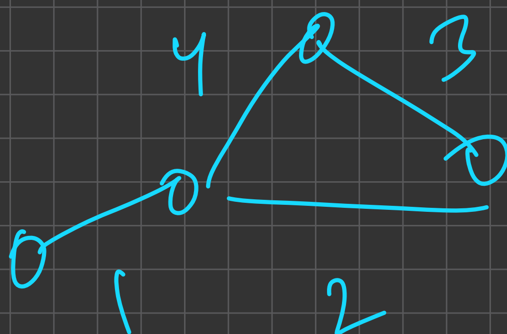
$$\left(\log 2 + \log 7 + \log 3 + \log 4 + \log 6\right) \longrightarrow$$

$$= \log\left(2 \times 7 \times 3 \times 4 - 6 \cdot - \right)$$









M.S.T B.L

Trees L.C.A

- Problem 1: Find the 2nd best minimum spanning tree of a Graph
- Problem 2: Village Water Distribution Problem

Prim's $\rightarrow$ $(n+m)\log n$ ✓

$\boxed{\text{Kruskal}}$ $\rightarrow$ $\underline{m\log m} + n$

$\underline{m >>> n}$

- Problem 1: Find the 2nd best minimum spanning tree of a Graph
- Problem 2: Village Water Distribution Problem