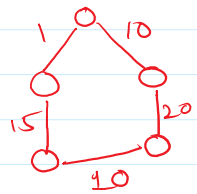
MST using Kruskal Algo (disjoint set)

01 February 2021 16:03

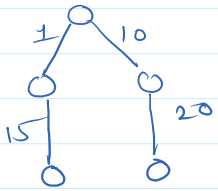
MST :- Minimum Spanning tree

Tree → connected & (V-1)

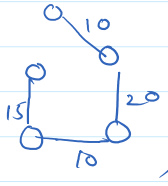
V = 5



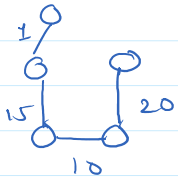
Spanning tree →



,

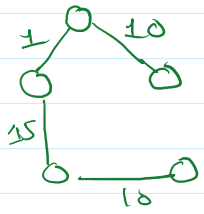


,



No of Spanning tree = $^5C_1$ = Total 5

Minimum cost ⇒ min edge weight sum →



cost = 1 + 10 + 15 + 10

36 ✓

We have 2 Algo

(i) Kruskal Algo → (work on Disjoint set)

(ii) Proms Algo → (heap)

Priority Queue → (heap)

Disjoint Set :- Sets that have nothing common DS.

$S_1 = \{1, 3, 5\}$ if $S_1 \cap S_2 = \phi$ (empty)

$S_2 = \{2, 4, 6\}$ then S_1 and S_2 are DisjointSet

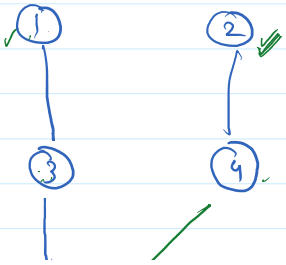
We can do two operation on Disjoint Sets:

(i) Find (n) n is an element

(ii) Union (n_1, n_2)

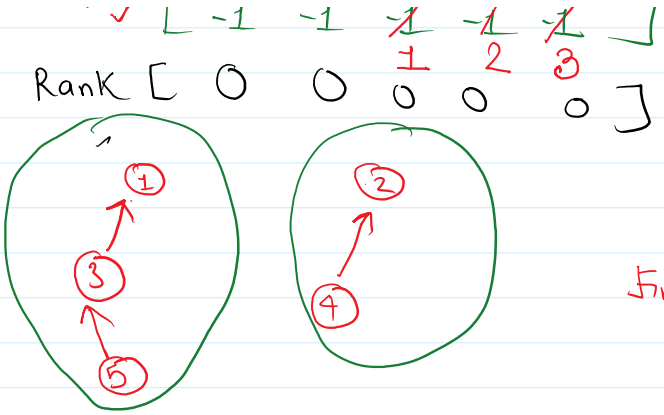
(i) Find(n)

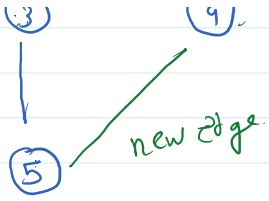
We will use parent array for Disjoint set



parent array

	1	2	3	4	5
	-1	-1	-1	-1	-1
Rank	0	0	1	2	3





Find (5) = return parent of all
= 1

Find (2) = 2

Find (4) = 2

Find (3) = 1

(ii) Union :— joining two set

How you know two element belong to diff set ?

if (parent are diff then Both belongs to different set.)

Union (a, b)

p_1 = find (a)

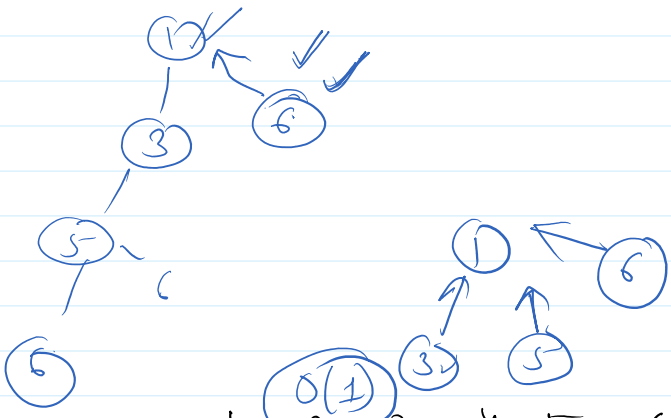
p_2 = find (b)

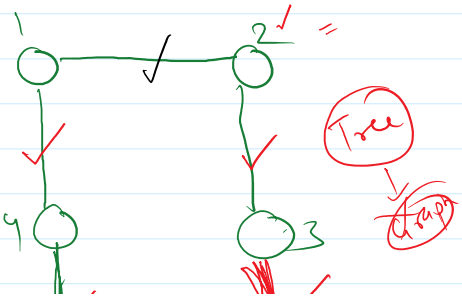
if (p_1 != p_2)

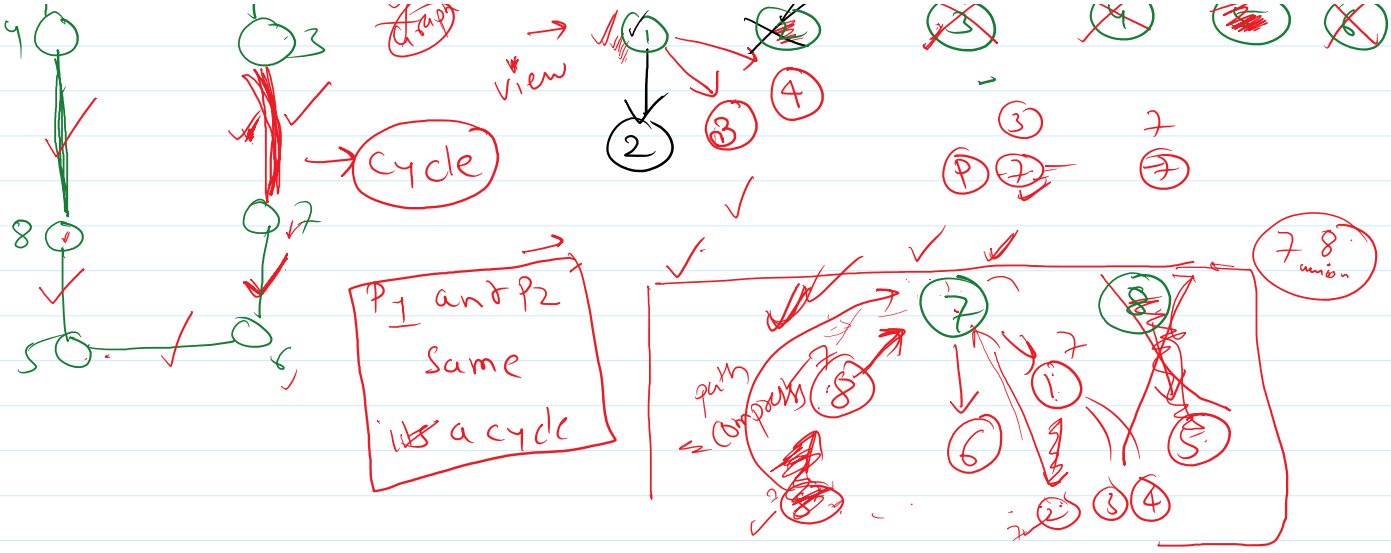
then Union it ✓ (higher Rank)

Rank :→ height of the parent ✓

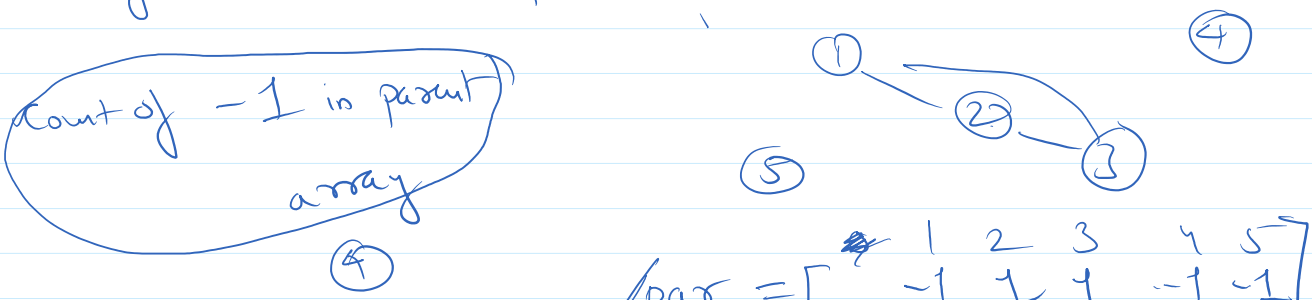
Path compression

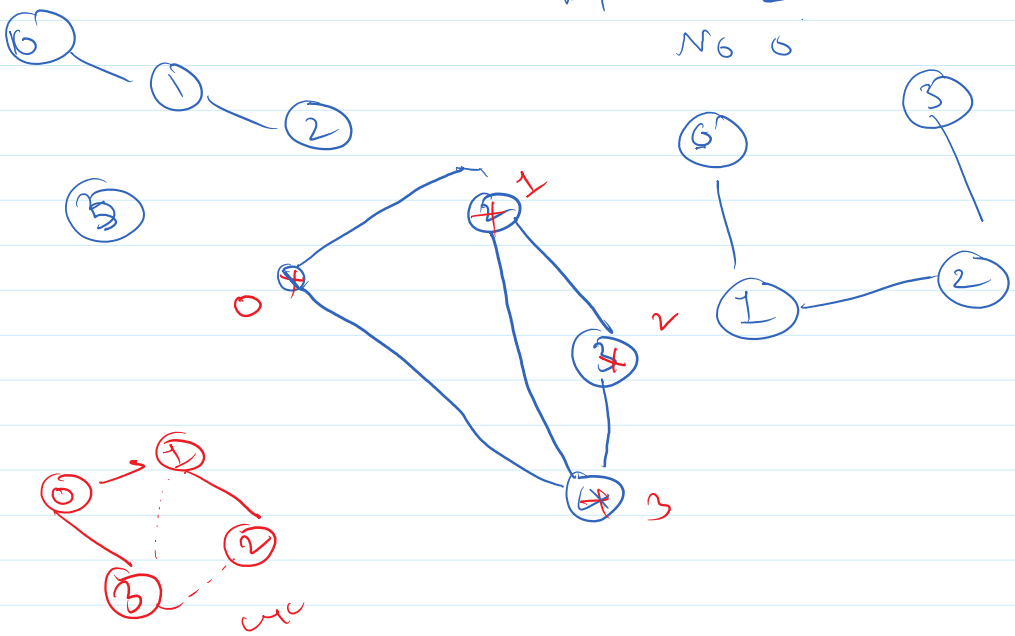






→ No of connected component ?

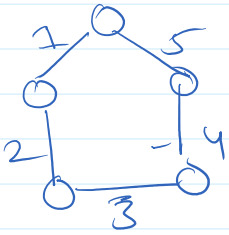


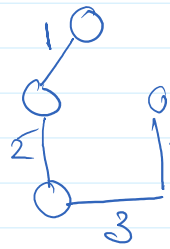


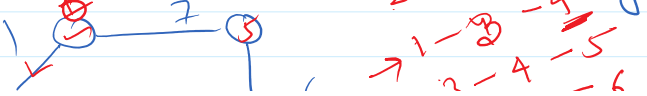
Kruskal → MST algo ✓

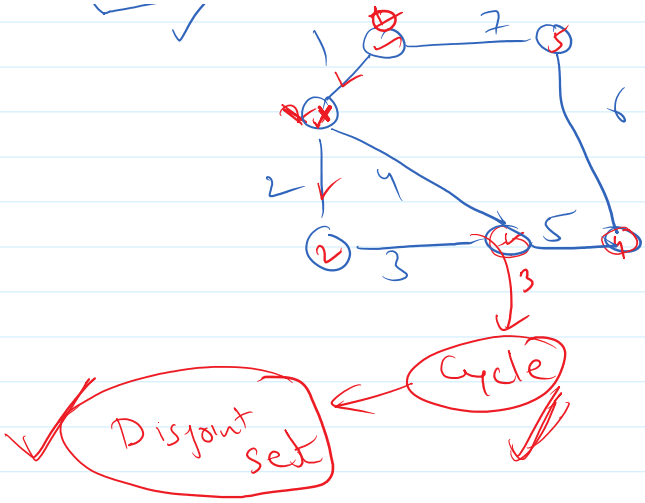
Sorted edges of graph → 7

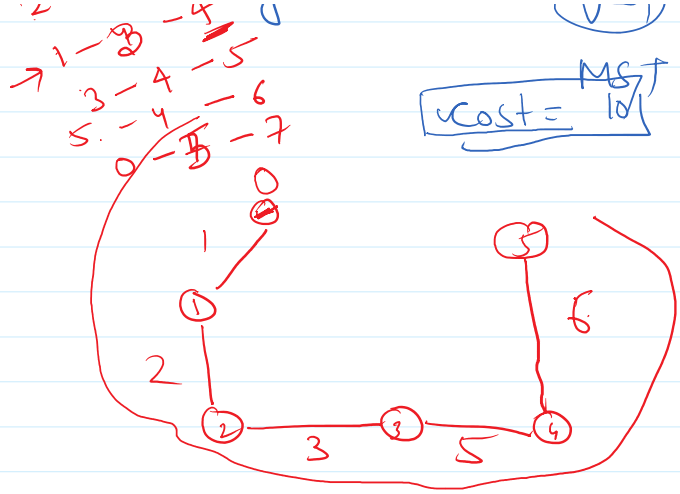
G ∈ G'











17

Kruskal algo

{

① Sort edges on weight

② while (edges < v-1)

{ if (Not cycle)

add edge e[i] to mst;

else ignore

}

cout << mst

}