Plan - Sunday

Saturday, 2 October 2021 10:32 PM

Morning

1. Number of Island-II
2. Redundant connection
3. Redundant connection. II

Evening

1. Kruskal Algo
2. Satisfiability of equality Eqn.
3. Sentence similarity

Number of Islands 2

Sunday, 3 October 2021 10:17 AM

Ex: 4x5 matrix

	0	1	2	3	4
0	0	0	~~0~~ 1	~~0~~ 1	0
1	0	0	~~0~~ 1	~~0~~ 1	~~0~~ 1
2	0	0	0	~~0~~ 1	0
3	0	0	0	0	0

no. of island on dynamic State

Array of Coordinate

0-3	1
1-2	2
0 2	1
1 4	2
2 3	3
1 3	1

Return → [1 2 1 2 3 1]

Dynamic States in Graph → DSU

Connected ⇒ Similarity ⇒ DSU concept

Brute force → process single coordinate and solve no. of island using DFS.

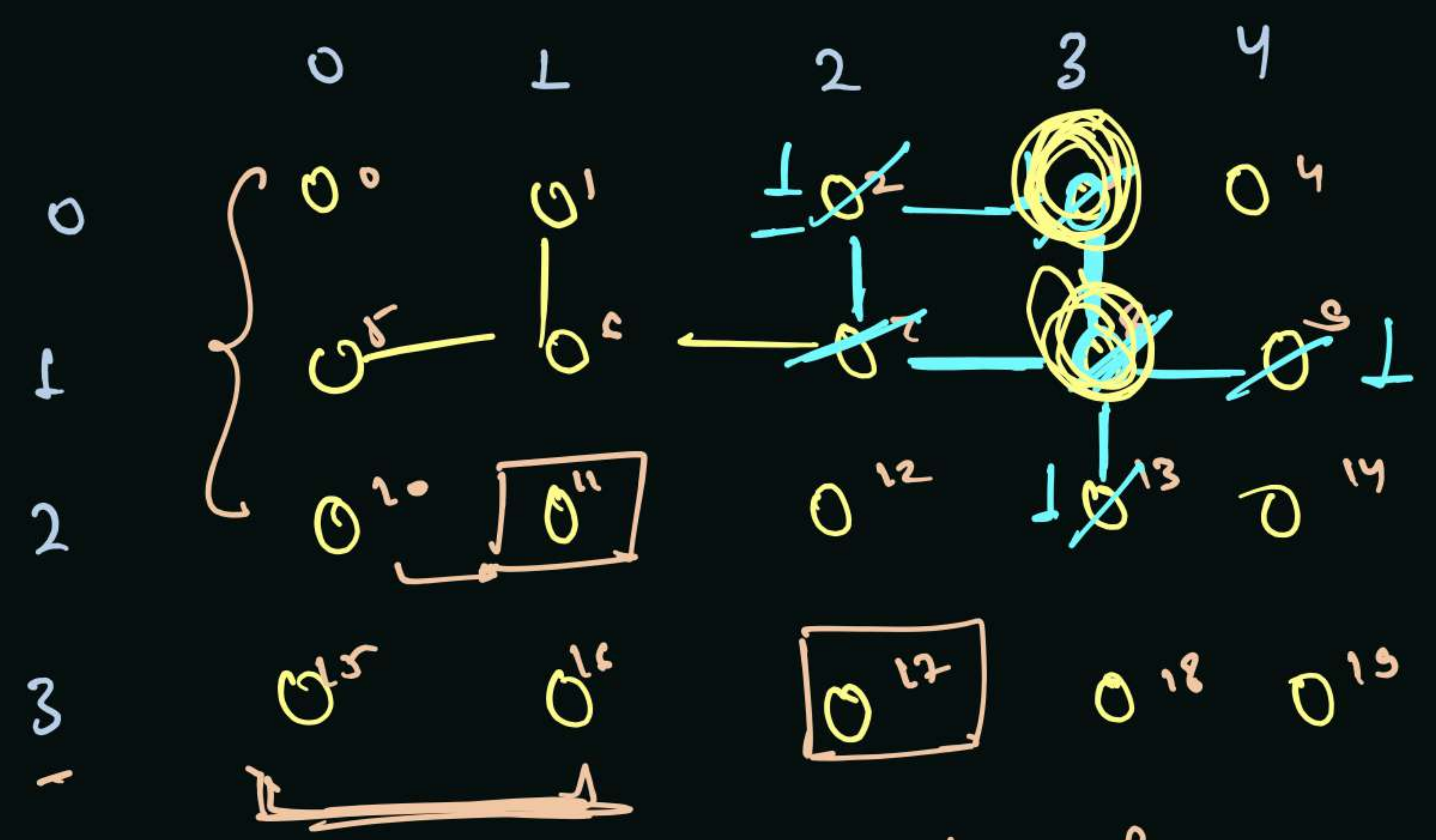
complexity = O[k * (V + E)]

k = no. of coordinates.

allowed = O(V + E + k) ⇒

(r, c) 2, 1 → coordinate box no →

Total cols * r + c = 5*2+1 = 11



Array of Coordinate

Not Edge ~~0-3~~

Not Edge	box No
0-3	3
1-2	7
0 2	2
1 4	9
2 3	13
1 3	8

// No. of Islands

count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~2~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~2~~ ~~2~~ 1

no. of islands

Conversion from Two D to oneD (Box no / cell No)

setleader

one-D array → boxNo → 0 1 2 4 5 6 10 11 12 14 15....

8 7 3 13 9

cellNo → 3 * 5 + 2 = 17

Conversion b/w 2D & 1D

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19

How to find box no.(1D) from cell No.(2D coordinate) →

coordinate (r, c)

$$\text{box No} = \underbrace{\text{total col} * r + c}$$

Ex: (3,1)

box No = 5 * 3 + 1 = 16

Ex (0,3)

box No = 5 * 0 + 3 = 3

Ex (2,4)

box No = 5 * 2 + 4 = 14

How to find cell No. from box No →

Ex: Total col = 5

$$\text{row} = \frac{\text{box No.}}{\text{Total Col}}$$

Col = box No % Total col

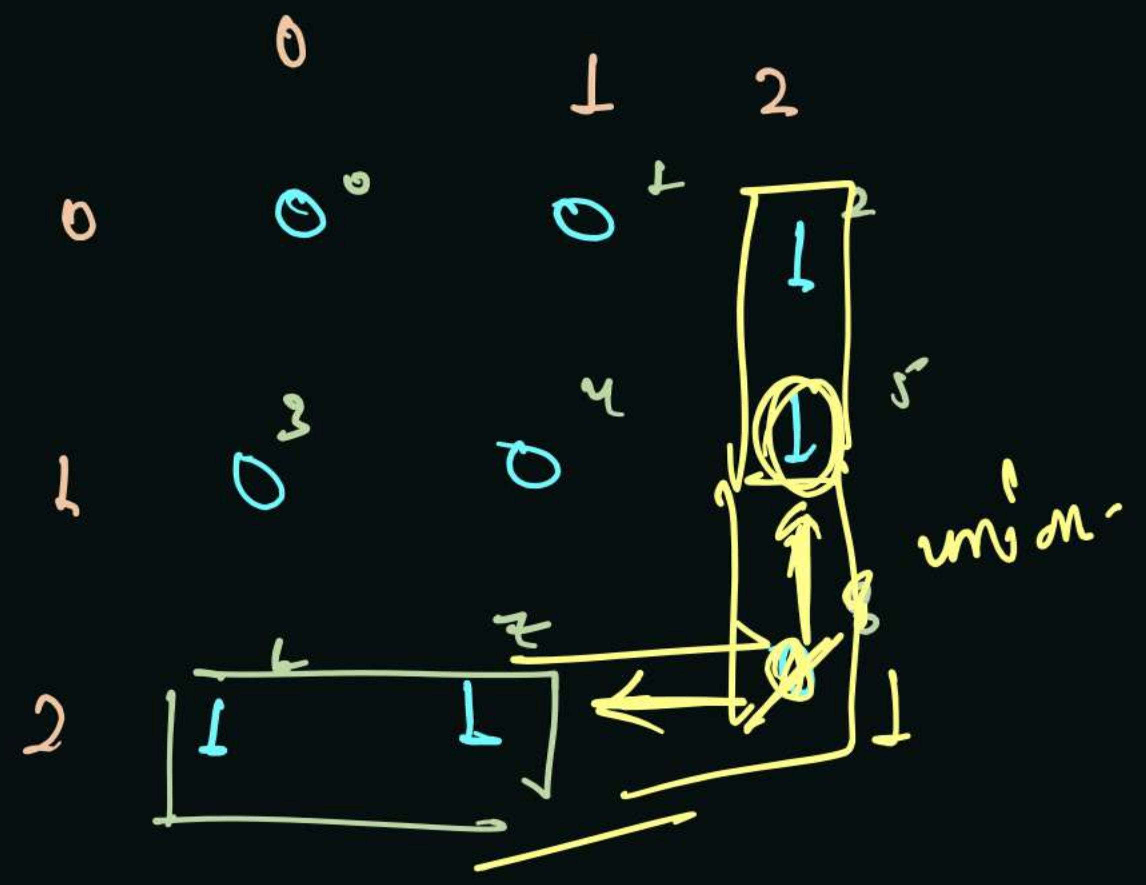
Ex: b.No = 16   row = $\frac{16}{5}$ = 3   (3,1)

Col 16 % 5 = 1

Ex. box.No = 3   row = $\frac{3}{5}$ = 0,   col = 3 % 5 = 3

(0,3)

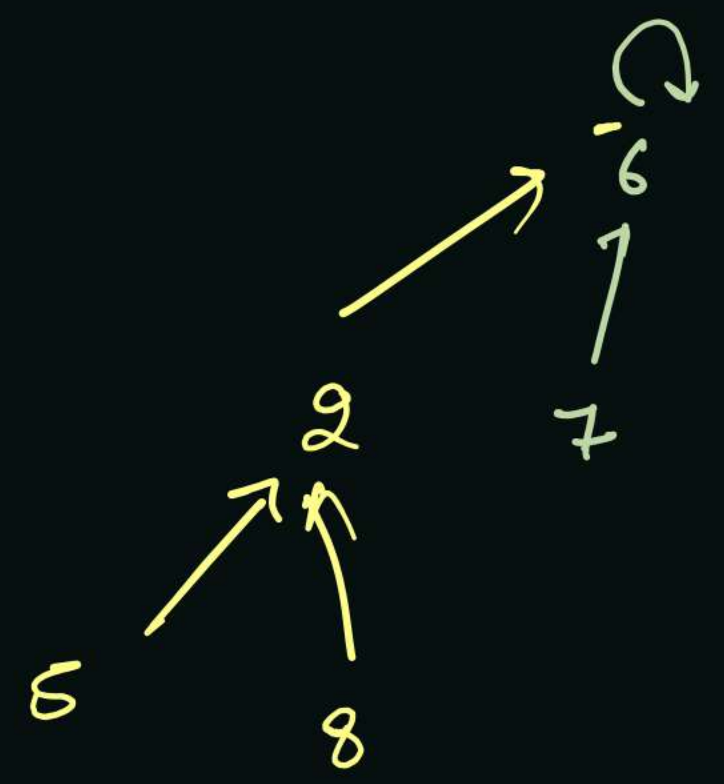
Ex = b.No - 14,   row = $\frac{14}{5}$ = 2,   col = 14 % 5 = 4   (2,4)



2-2 → 1

count = ~~2~~ ~~2~~ ~~2~~ 1

t l d r



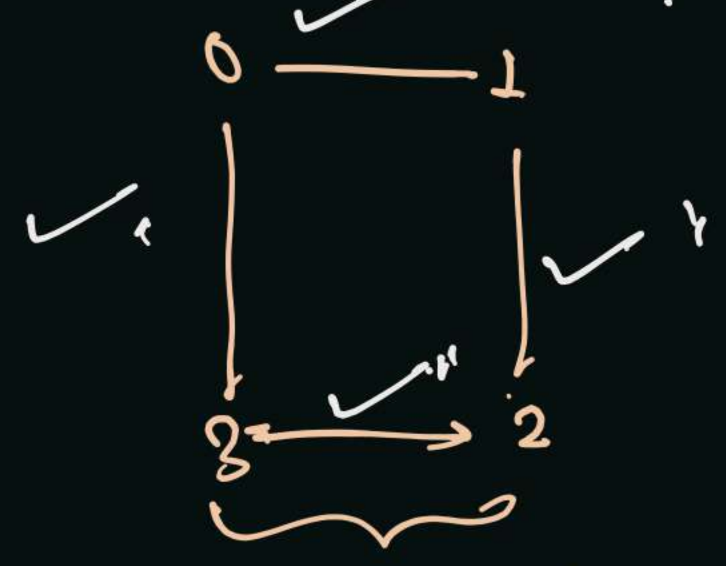
Redundant Connection

Sunday, 3 October 2021 12:18 PM

Given a tree with addition one more Edge, tree convert into graph, find that redundant find,

* A graph is a tree if Acyclic and Connected.

valid Removal for con



convert in graph from Tree.

NOTE: There must be an Edge present from Removal of that Edge Graph convert into tree.

order

0 – 1

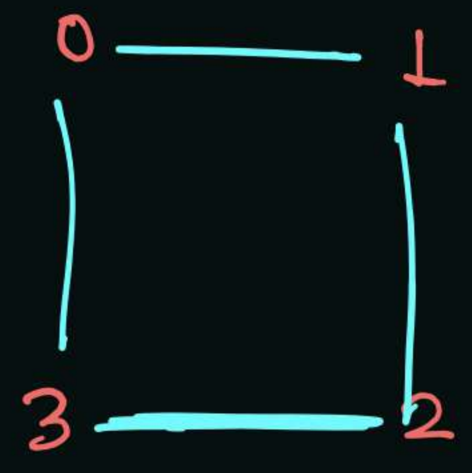
0 – 3

1 – 2

$\boxed{2 – 3}$

this Edge is responsible for conversion so it is Result.

⇒ so if there are multiple possibility of removal of Edge then remove last Edge provided order.



if a tree have 'e' edgy then it must have 'e+1' vertices.

0 – 1 ✓

1 – 2 ✓

0 – 3 ✓

$\boxed{2 – 3}$

2 have set leader = 0

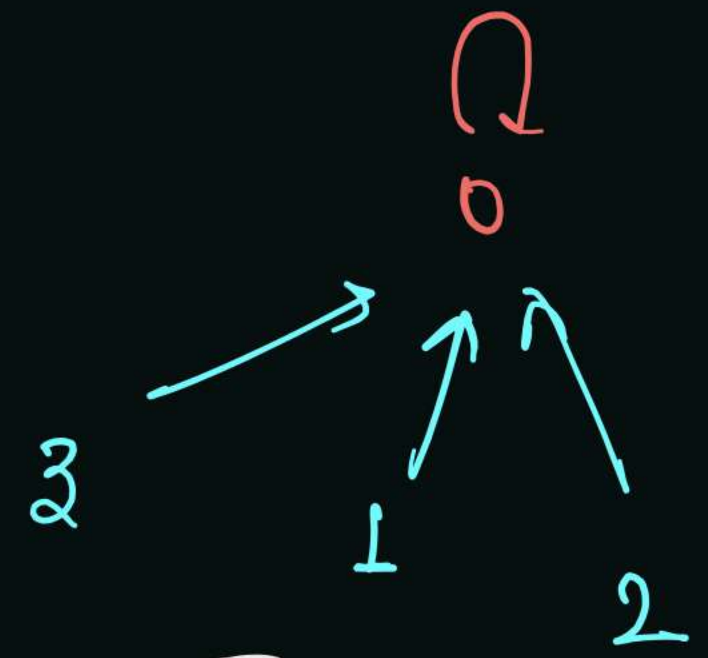
3 have set leader = 0

Tasted

DSU

if same set leader exists that mean this is cyclic Edge.

No. of vertex → parent



there must be single cycle, and it is true that current Edge is responsible Edge.

Similar Grouping, single cycle

Merge set

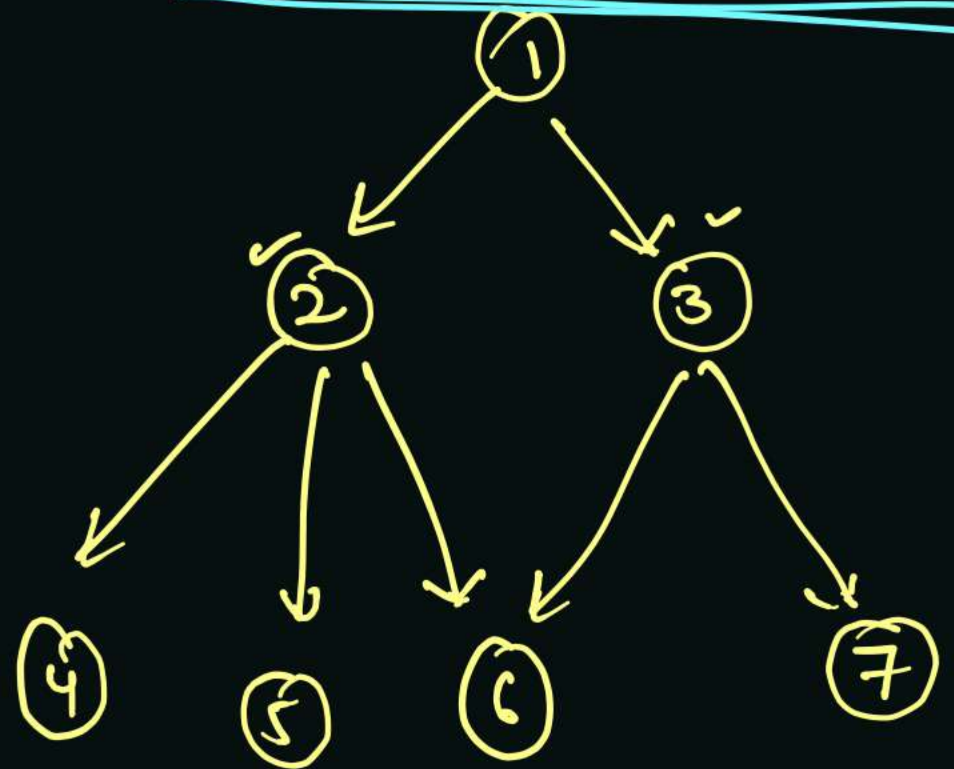
Redundant Connection 2

Sunday, 3 October 2021 1:06 PM

1- based Index, directed acyclic and connected graph (initially)
add one more Edge and tree convert into graph. find redundant connection from removal of that Edge, graph convert into tree.
last possible Edge is Redundant.

graph is not a tree →
✓→ connected
✓→ Acyclic
✓→ vertex have single parent

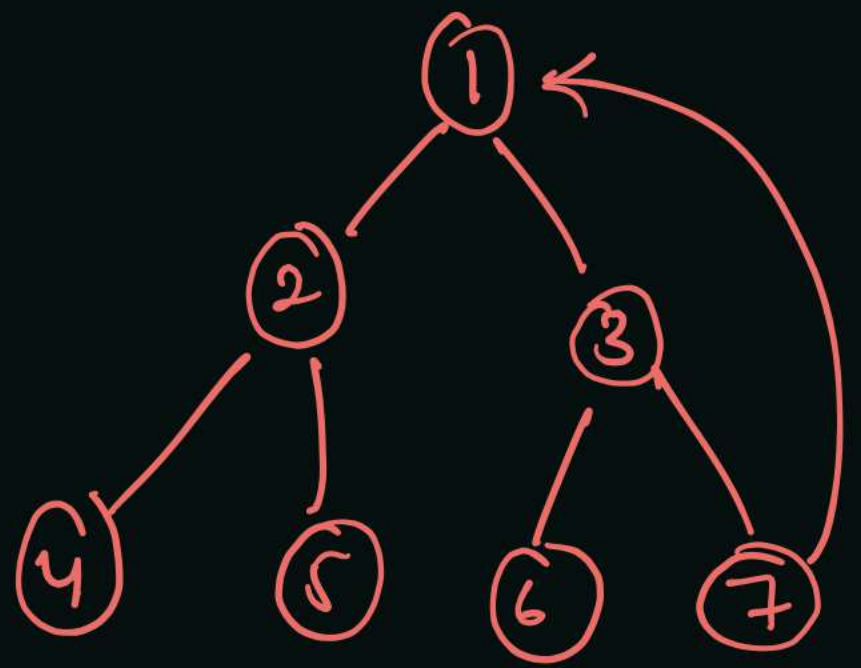
Case - I - Two parent



connected ✓
Acyclic ✓
Single parent ✗

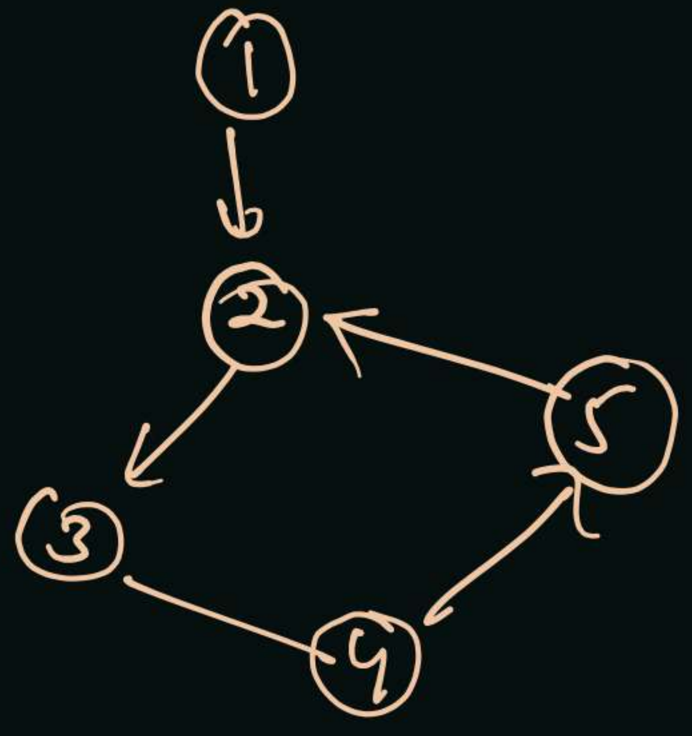
Not a tree.

Case-II Single parent Cyclic

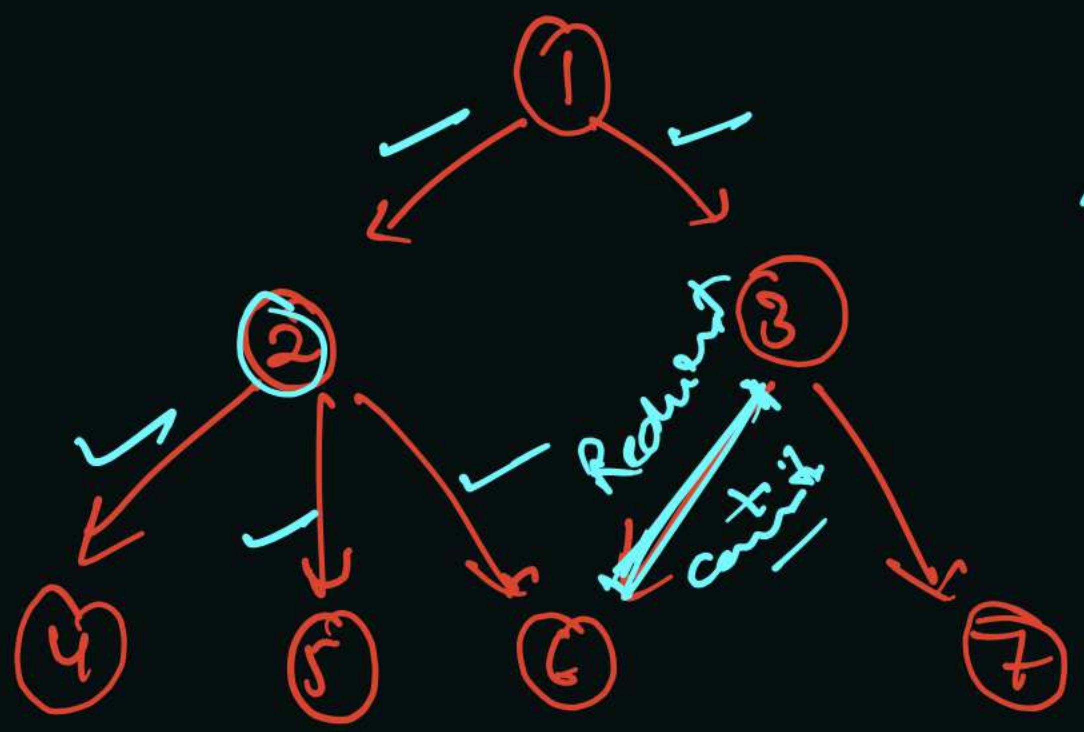


Case-III

two parent and Cyclic



Case - I   Two parent for a vertex



order of Edge

root → 0. 1→2
1. 1→3
2. 2→4
3. 2→5
4. 2→6
5. $\boxed{3\to 6}$
6. 3→7

} given array

Result because it create → 1
two parent of 6 ⇒ degree 2

Indegree - concept

No. of neighbours pointing toward a vertex is degree of that vertex.

Make a degree array and initialise it as '-1'

1-2
u v

Index i

indeg[v] = i

	1	2	3	4	5	6	7
	-1	~~-1~~	~~-1~~	~~-1~~	~~-1~~	~~-1~~	-1
rootEdge Index →		0	1	2	3	$\boxed{4}$	

→ that single Extra Edge is here.

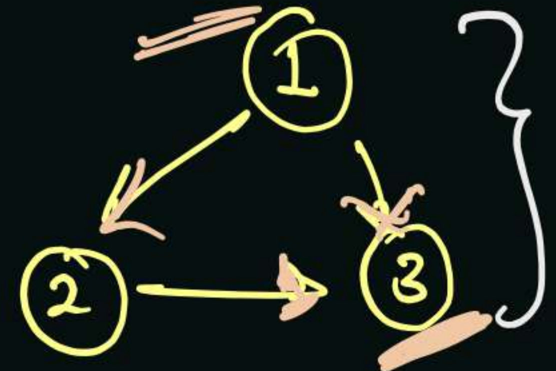
if in Edge of u, v

indegree[v] is not -1 that means there is already another parent of v is available.

★ Now In case-II [graph is cyclic] → for surety of cycle check indegree first

NOTE: If we have directed graph and we are trying to find if it is cyclic or not then DSU is not enough to find this.

cross Example.

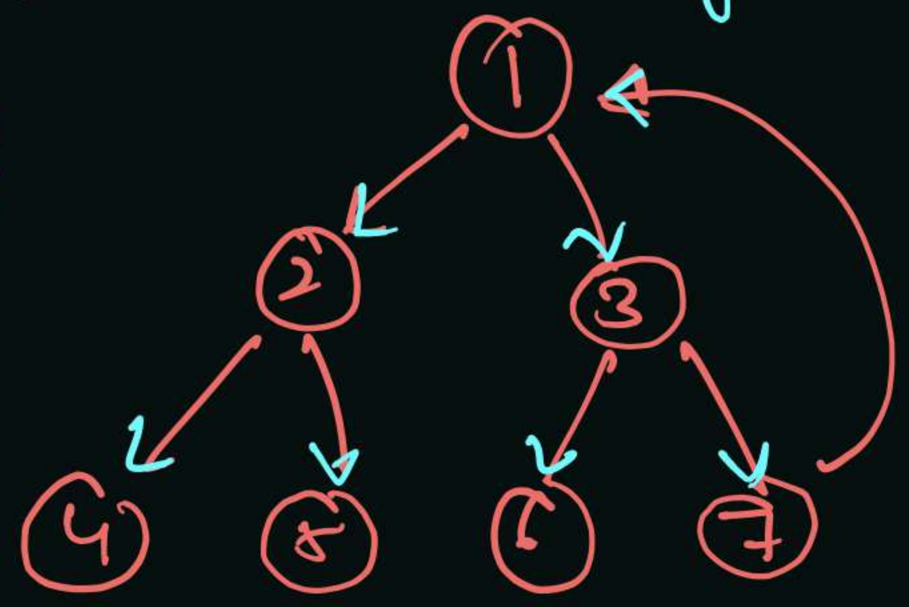
⟹ 

DSU says it is cyclic

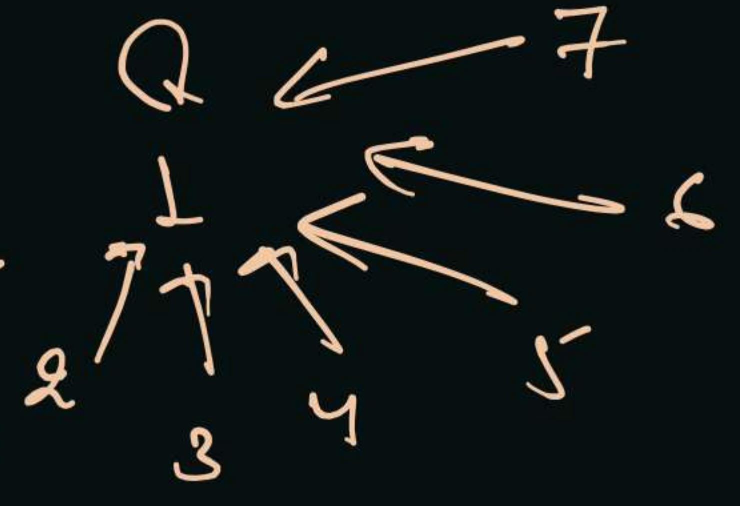
But it is not!

1 2
2 3
1 3 → have same set leader

But if in directed graph we know that there must be a cycle and single parent for all vertices. and also one one redundant edge is creating that cycle then we can find that edge using DSU concept.

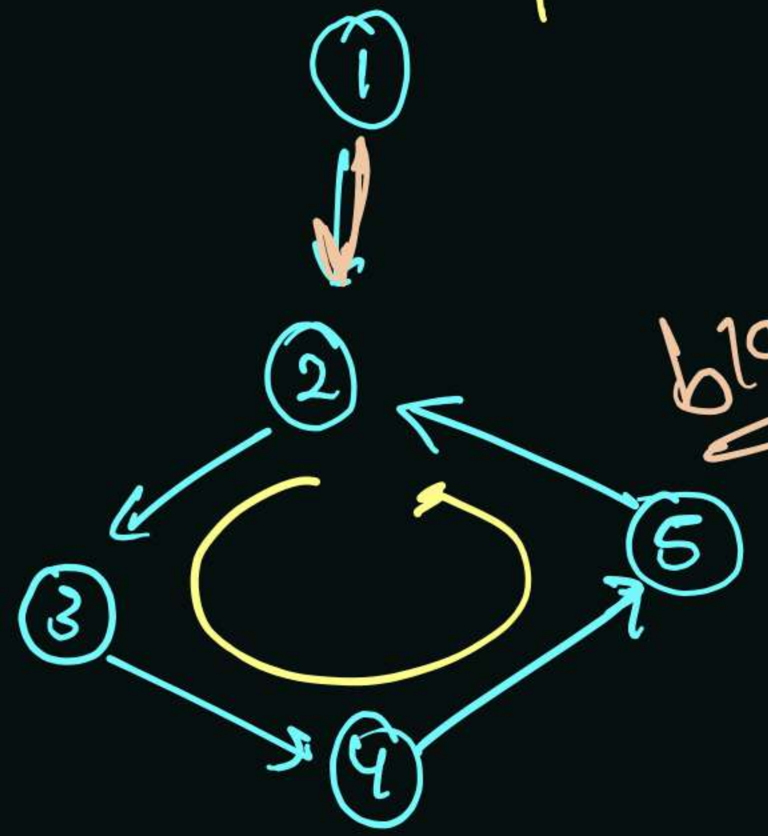


1 2 ✓
1 3 ✓
2 4 ✓
2 5 ✓
3 6 ✓
3 7 ✓
7 1 ✓ ⟶ Both have same set leader



7-1 is ans

Case-III [cyclic and two parent] order of Edge



	0	2-3
	1	3-4
blacklist	2	4-5
	3	5-2
	4	1-2 Result

Indegree

1	2	3	4	5
-1	~~-1~~	~~-1~~	~~-1~~	~~-1~~
	2	0	1	2

$\boxed{1-2}$ → Redent Connection

After Indegree mangement, we can see that 1-2 is Redmdent connection but After removing (1-2) it still not a tree, that means we have to manage these cases in different ways.

Point out blacklist Edge

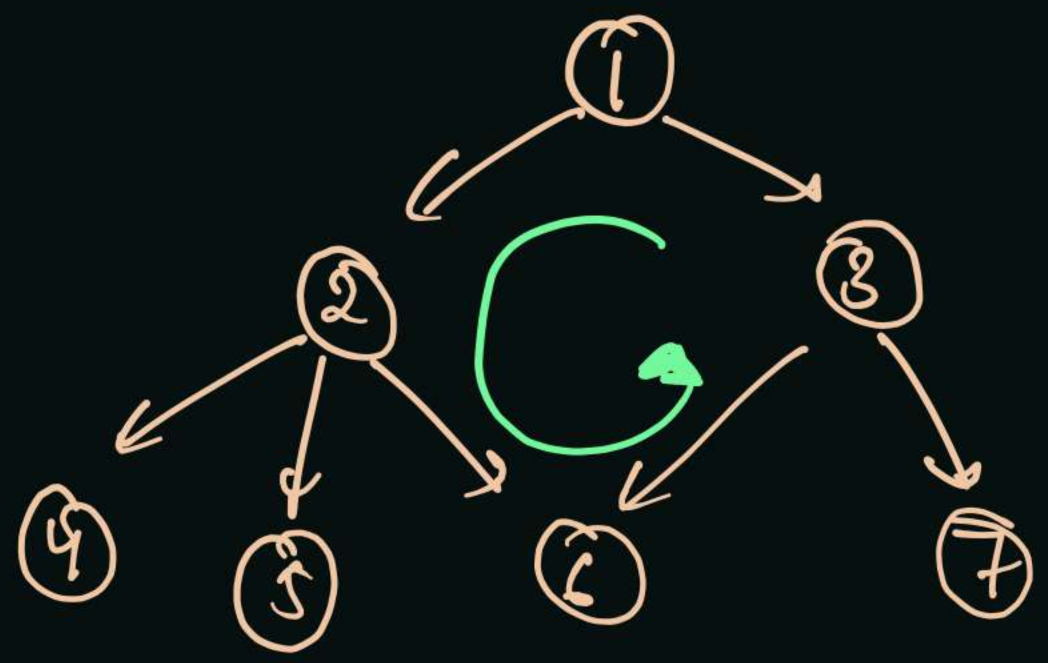
blacklist1 = ~~-1~~ 3

blacklist2 = ~~-1~~ 4

we are not sure that if there is cycle or cycle with two parent or two parent

for sureity of cycle, we have to make sure that if it has a vertex have two parent or not.

ignore bl1 Edge and check cycle and same for (2).



0. 1→2 ✓
1. 1→3 ✓
2. 2→4 ✓
3. 2→5 ✓
4. 2→6 ✓
5. 3→6 ignore
6. 3→7

Conclusion: if blacklist 1 == -1

then that means no vertex have two parent and there must be a cycle

	0	1	2	3	4	5	6	7
Indegree	-1	-1	~~-1~~ 0	~~-1~~ 1	~~-1~~ 2	~~-1~~ 3	~~-1~~ 4	-1

it helps to identify two Edges

- blacklist 1 = ~~-1~~ 5
- blacklist 2 = ~~-1~~ 4

ignore blist 1 Edge and check if graph from remaining Edge is cyclic or not → using DSU

→ Cyclic → blacklist 2 Edge

→ Acyclic → blacklist 1 ✓