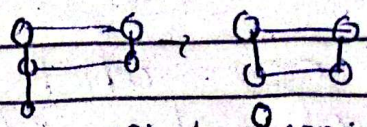
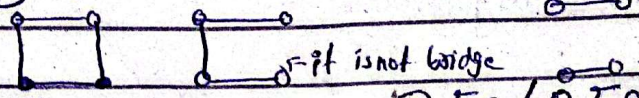
→ What is Bridge:- ex:-



in graph theory, a bridge is a single connection. (an edge), if you remove it, it would break the graph into two or more separate pieces. ex



→ what does "Using DFS/BFS to find Bridges" mean??

it means we use the DFS or BFS algorithm as a powerful tool to walk through the graph and identify which edges are the critical bridges.

Think of it like this:-

- The graph is the city you need to explore.
- DFS/BFS is your method of exploration (walking down every street)
- Finding Bridges is the specific mission you are on (to find which bridges are critical)

→ so which algorithm will be used for finding the bridge in the graphs.

so we will use some logic in the DFS algorithm.

→ pre-req you must have the knowledge of DFS.

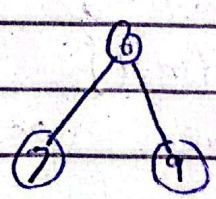
→ so for finding the bridges in the graph, what we will do? we will take two seprate arrays.

1, one is the time of insertion. (DFS time insertion)

2, lowest time of insertion.

(min) lowest time of insertion of all adjacent nodes apart from parent.

ex:-

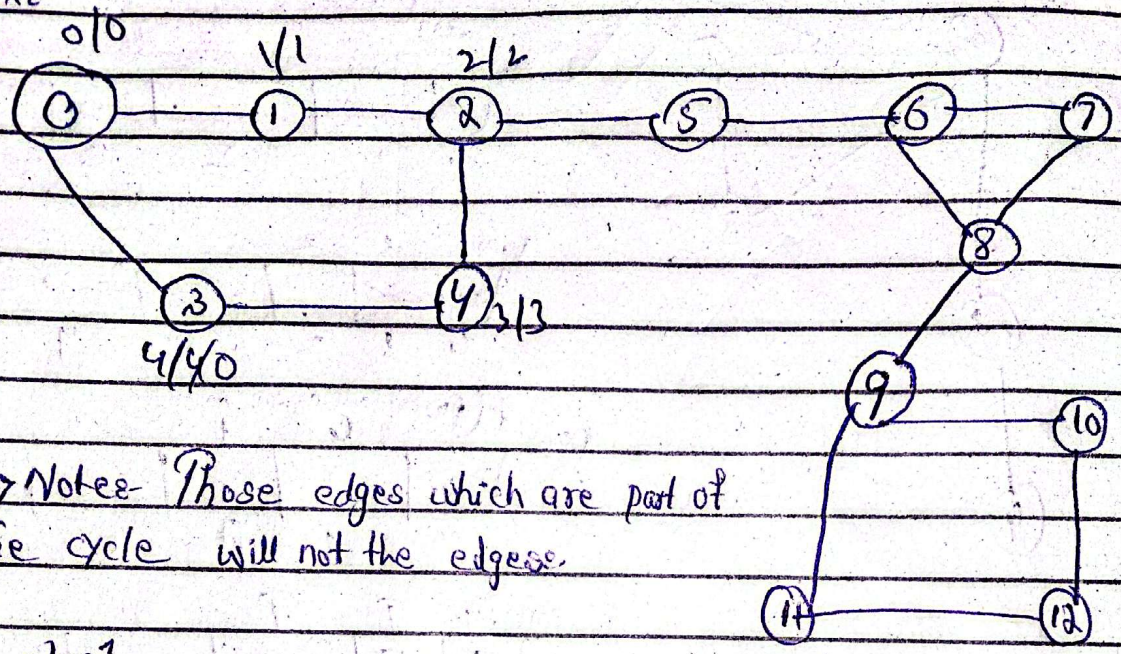


so t[]

low[]

→ Time of insertion is the first step.

exe-



→ Notes- Those edges which are part of the cycle will not the edgese.

step1

→ for Bridg exist condition:

means neighbour.

discovery [parent] < low [child]

means node

so in this case bridge will existo

→ step2: condition for not a bridge:

disc [parent] >= low [child]

node

neighb.

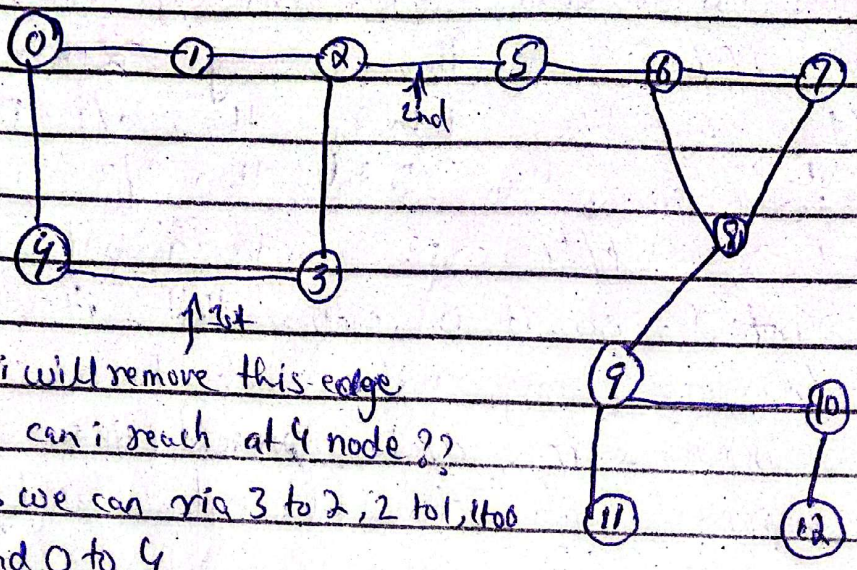
so first we visit to zero so it's discovery time and lower time 0/0, so then we go ~~Zero~~ to 1 the 1/1 and the 1 to 2 the 2/2 Discover and low time, then 2 to 4 the 3/3 D and L Time then 4 to 3 → 4/4 so 3 to 0 but 0 is already visited, so it is confirm that it will not bridge because if any node visit twice, so there cycle will occur, so if cycle present then it is confirm, that it should not bridge.

So for 4/4, we will copy minimum.
4/0

and then we go back 3 to 4 and in 4 node 3/3 the copy minimum 3/0

disc of [node] ≤ low [neig]
3 ≤ 0

example start from this



if i will remove this edge
so can i reach at 4 node??
Yes we can via 3 to 2, 2 to 1, 1 to 0
and 0 to 4.
so this will not consider as a bridge.

and if i will remove the 2 to 5 then ic can not reach the nod 5.
so it is bridge.

Same for 8 to 9 it is also bridge.
so we have to find how many bridges are present.

→ so how can we solve this problem first is brute force algorithm.
in brute force algorth we remove every edge one by one and then check can we reach at or not.
so if we reach then it will not consider as a bridge.
brute force time complexity will remain very high.

1 → Note all the edges which are part of the cycle, will never be as a bridge.

2 Note 2: A Node which is already visited will never be a bridge.

~~ex~~

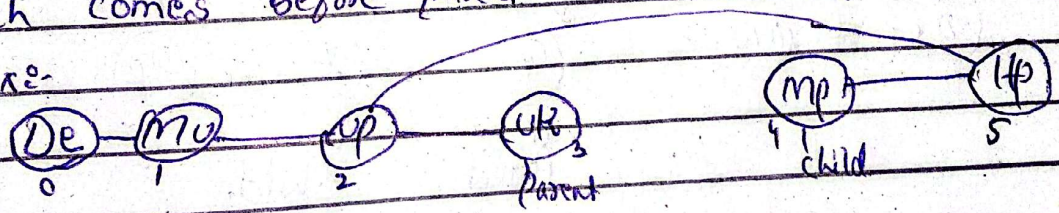
→ Condition for not a bridge:- There should not be a bridge in between parent and child.

ex:-

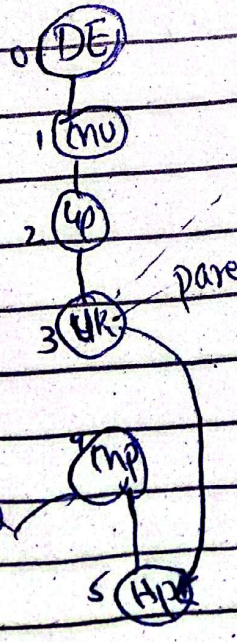
Parent → child.

→ If from a child, if we can visit any node which comes to parent or the node which comes before parent, it will not a bridge.

ex:-



so



→ so yha UK pooch rha hai MP sai apny child sai kia tum kisi b node sai/ko visit kar sakty ho jo jo UK(3) ya us sai pehly aye ho.

MP=4

HP=5

UP=2.

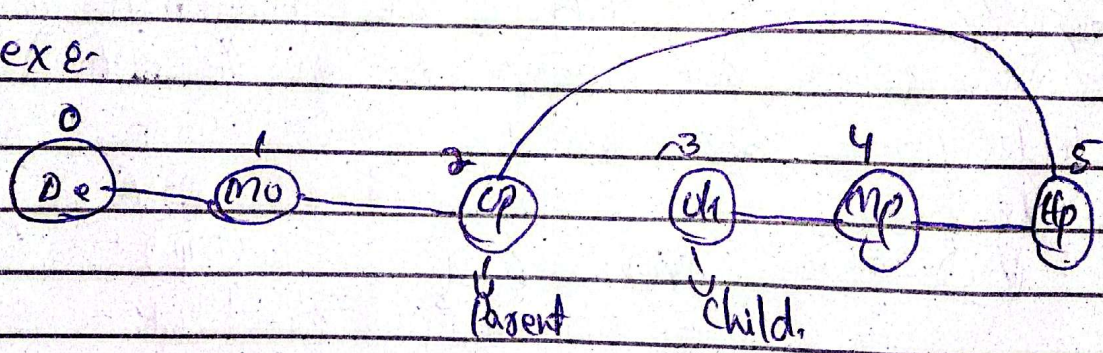
so if koe be value 3 ya us sai choti ho jaata hai to wheapai bridge exist nh krty ga.

simple: if any value here, is less than or equal to parent value, it is not a bridge.

so simple: if parent ≥ child value then not a bridge. or (agr) if child ≤ parent then not a bridge.

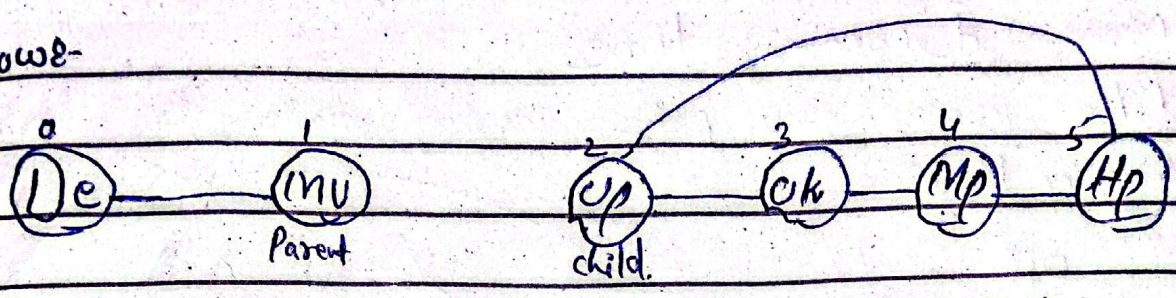
→ iso means agr hum child wali agr hum parent node ya us wali pehly wala node visit kar sakty hai to it will not a bride.

exp:

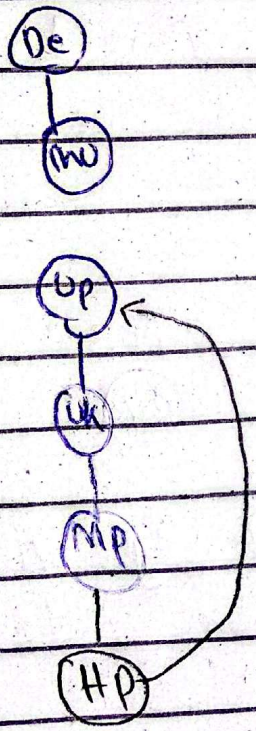


iso here we can go to child to parent and before parent nodes then it is not a bridge.

Crowe-



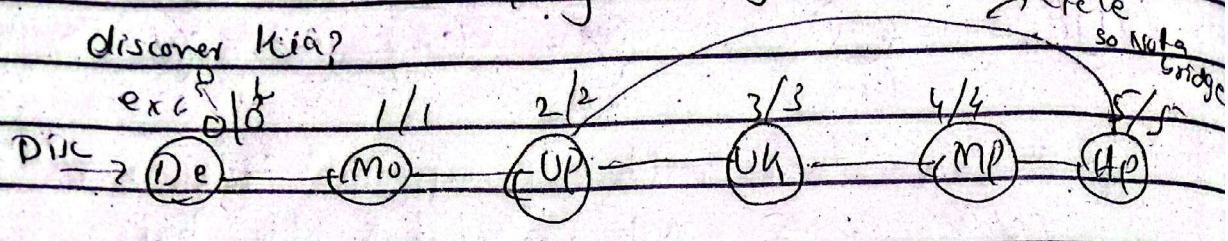
→ so kia hum child sai hum parent Node ya us sa pehly wala node visit kar sakty hai.
to answer is no.
so it is the Bridge.



if child greater than parent then it is a bridge.
→ child > parent = bridge.
2 > 1 = bridge.

→ Now:- Discovery time

Like wab wai pehy meny node ko kab discover kia?

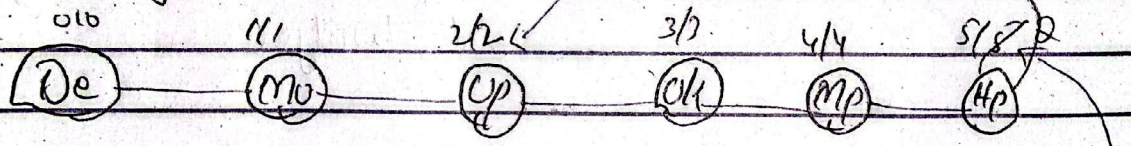


→ Low :- Minimum possible time to discover a node which we can visit without using parent.

Lowest

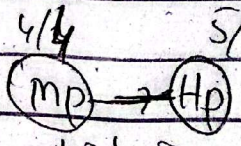
Now:

already visited:



so the node which is already visited then we will copy the low/minimum to that node, so 2 is the minimum.

Now hp to Mp back check



is this bridge? so from both which is minimum copy it into the Mp.

so Mp is parent and Hp is child.

the condition of bridge is:

child > parent

or

disc [parent] > low [child]

then it will be bridge.

so disc of parent = 4

low of child = 2

→ For Not a bridge.

parent > child.

4 > child = 2

4 > 2 Hence Not a bridge.

same for others :-