Binary Tree Playlist...

Leetcode -2385

~~Medium~~

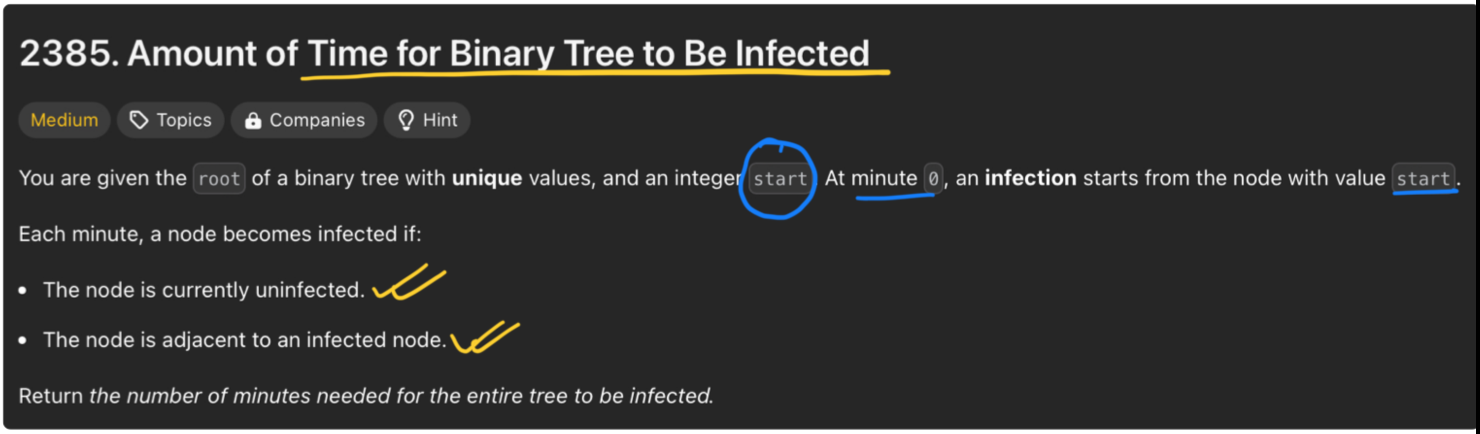
Video - 34

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK →

Approach-1



2385. Amount of Time for Binary Tree to Be Infected

You are given the `root` of a binary tree with **unique** values, and an integer `start`. At minute `0`, an **infection** starts from the node with value `start`.

Each minute, a node becomes infected if:

- The node is currently uninfected.
- The node is adjacent to an infected node.

Return *the number of minutes needed for the entire tree to be infected.*

Example :-

unique values

start = 3

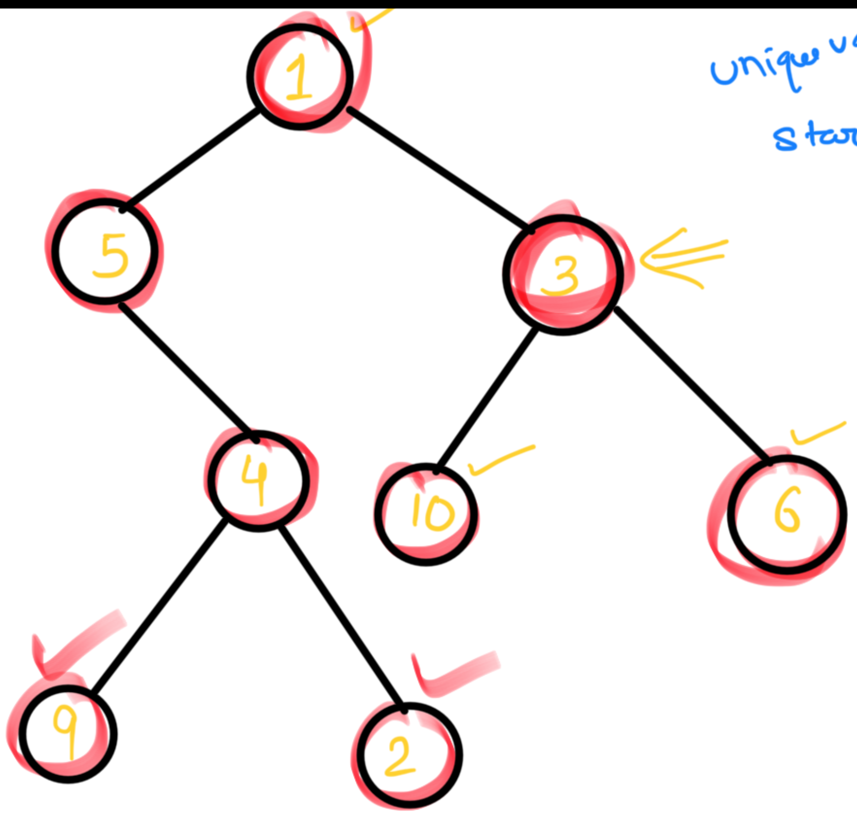
$\boxed{3}$ → minute = 0

1, 10, 6 → minut = 1

5 → mi = 2

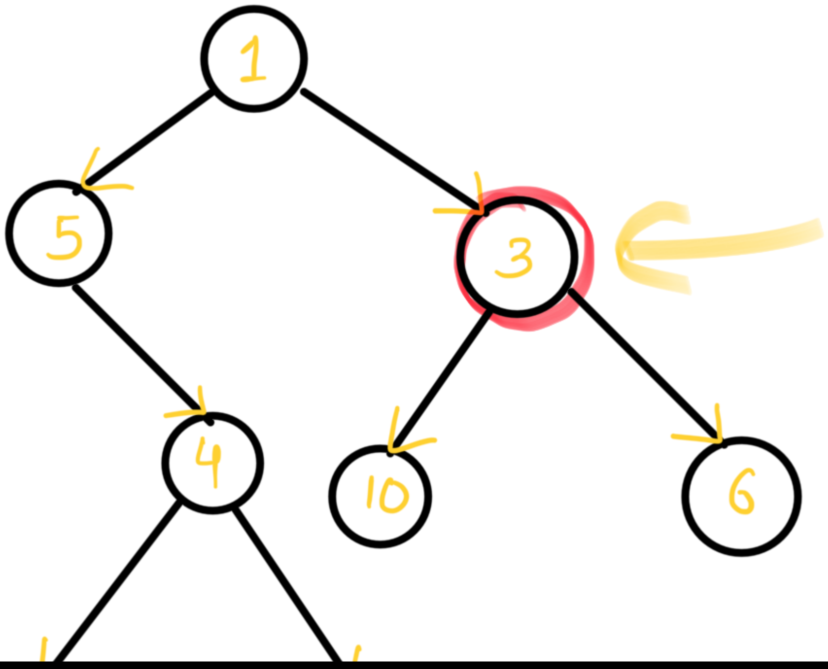
4 → mi = 3

9, 2 → m = $\boxed{4}$



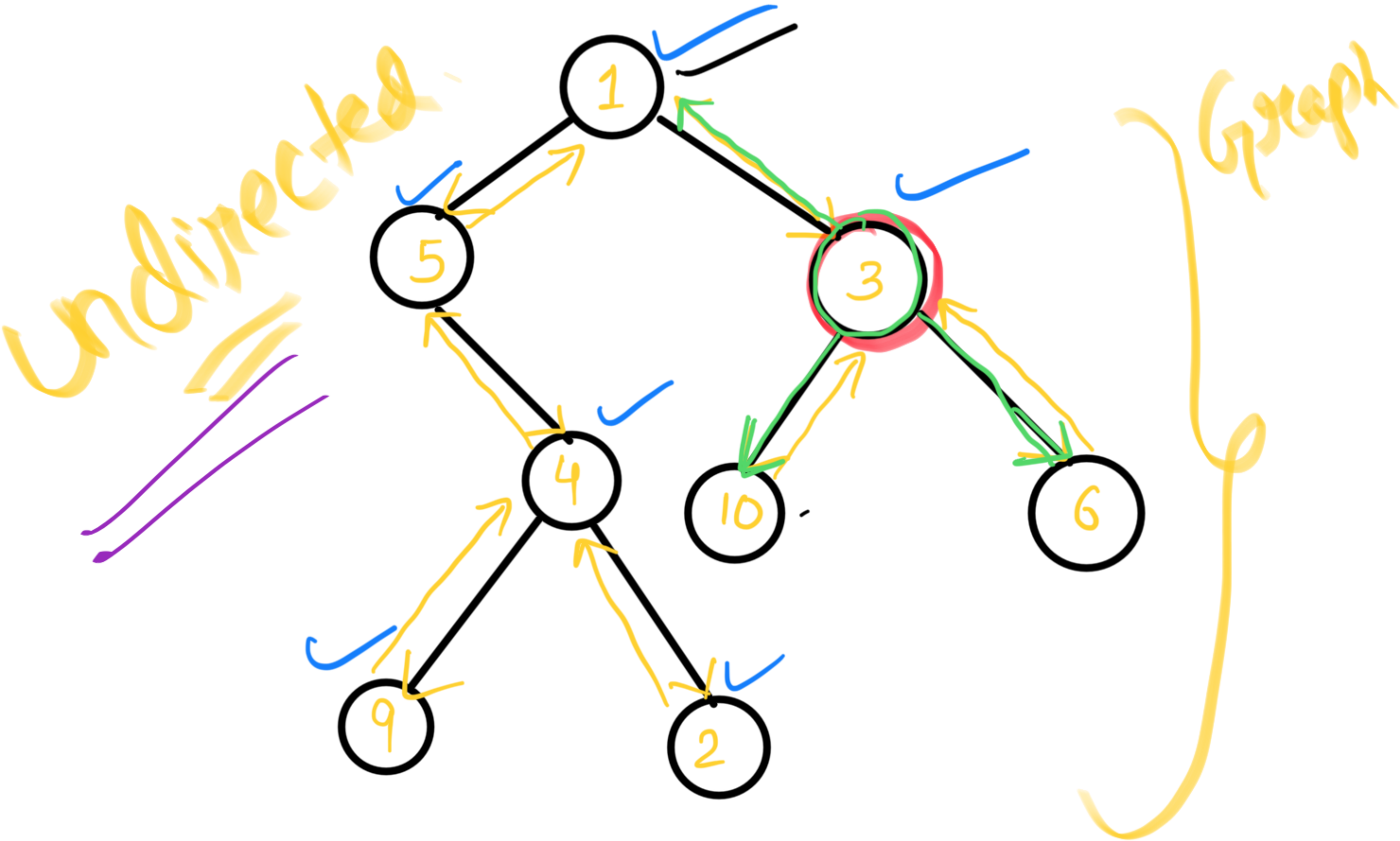
minute = 4

Intuition :-



1. How to traverse back???

2. BFS injection...



```
unordered_map<int, vector<int>> adj ;
```

1 → 5, 3

5 → 1 4

4 → ~~5~~, 9, 2

9 → ~~4~~

2 → ~~4~~

3 → 1, 10, 6

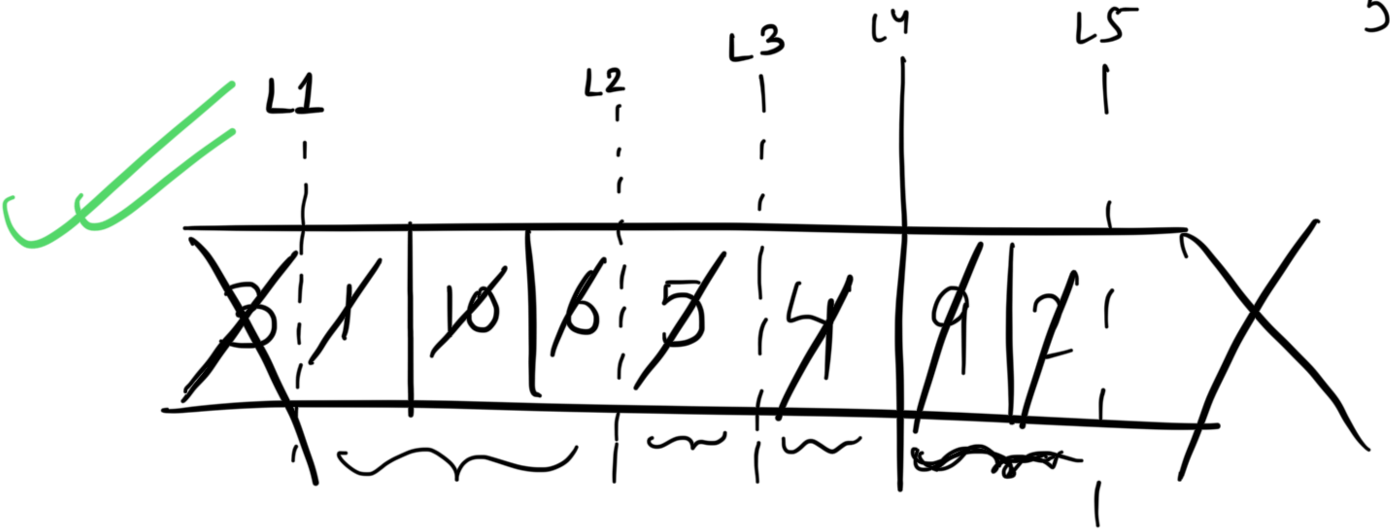
10 → 3

6 → 3

visited

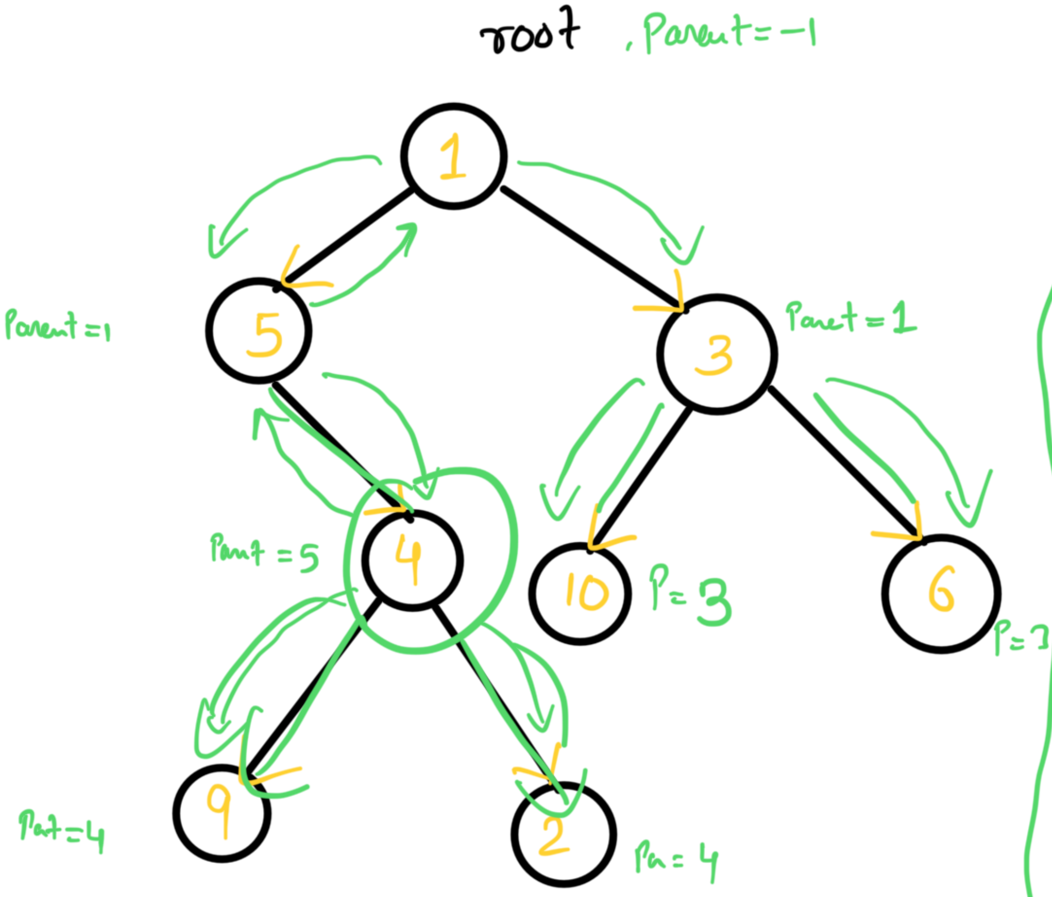
3, 1, 10, 6, 5
4

minute = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5



ret (minute -1) ;

Forming Graph:- (adj)



adj :-

int → vector<int>

1 → 5, 3

5 → 1, 4

4 → 5, 9, 2

9 → 4

2 → 4

3 → 1, 10, 6

10 → 3

6 → 3

```
makeGoraph(adj, parent, curr) {

    if (Parent != -1) { // Curr ka parent
        adj[curr->val].push_back(Pant);
    }
    if( curr->left)
        adj[curr->val].push_back(curr->left->val);
```

```
            if (curr->right)
                  adj[curr->val].pushback(curr->right->val);


            makeGraph(adj, curr->val, curr->next);
            makeGr   (adj, curr->val, curr->right);

    }
```