Problem: A company has n employees with unique ID's from 0 - n-1. The head of the company has the ID headID

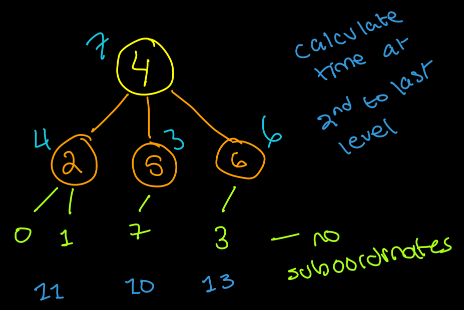
0 1 2 3 4 5 6 7

headID = 4 (headID's #)

Receive managers array where managers[i] is the manager's ID for employee i.
Each employee has 1 direct manager.
Company head has no manager: so managers[headID] = -1.
Guaranteed subordination relationships will have a tree structure

head

8 employees: 0 1 2 3 4 5 6 7 headID = 4
managers : 2 2 4 6 -1 4 4 5



N-ary trees

—

Graph Questions

convert to 2D array

inform subs about news

informTime[i]

employee i to inform direct subs

RETURN # of min takes to inform employees

0	1	2	3	4	5	6	7
0	0	4	0	7	3	6	0

cost array

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

0	
1	
2	0 1
3	
4	2 5 6
5	7
6	3
7	

1

adj. list

indices + managers

perform dfs on head manager's neighbors

Intro:
- Verify Constraints
- Create Testcases

Brute Force:
- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity

Optimal:
- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases

- Analyze time and space complexity

max_time=0;

empty []s ↓ # of rows = # of employees = n

1st: convert to 2D Adj. List:

adj_list = [ ]

iterate thru emp ids:

if emp_id != headID

adj_list [[manager[emp_id]]]

0

manager[0] = 2

.append(emp_id)

2nd call dfs

```
iterate thru head's neighbors
For neighbor in adj_list[headID]:
    new_time = informTime[headID]
    dfs(adj_list, informTime, new_time, neighbor)
    max_time = max(max_time, new_time)

return max_time
```

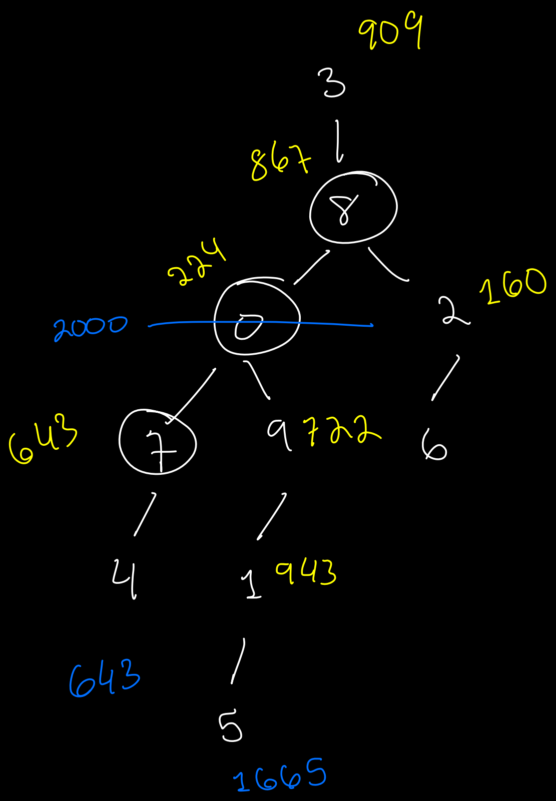
```
dfs(vertex, adj_list, informTime, new_time):
    if informTime[vertex] == 0
        return

    new_time += informTime[vertex]
    for neighbor in adj_list[vertex]:
        dfs(neighbor, adj_list, informTime,
                                 new_time)
```

adj. list

0	8
1	9
2	8
$\boxed{3}$	-1
4	7
5	1
6	2
7	0
8	3
9	0

⇒

0	7, 9
1	5
2	6
3	8
4	
5	
6	
7	4
8	0, 2
9	1


909
3
867
8
224
2000
0
2
160
643
7
9
722
6
4
1
943
643
5
1665