

{% note info %} **摘要** Title: 285. 没有上司的舞会 Tag: 树形dp、dp Memory Limit: 64 MB Time Limit: 1000 ms  
{% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

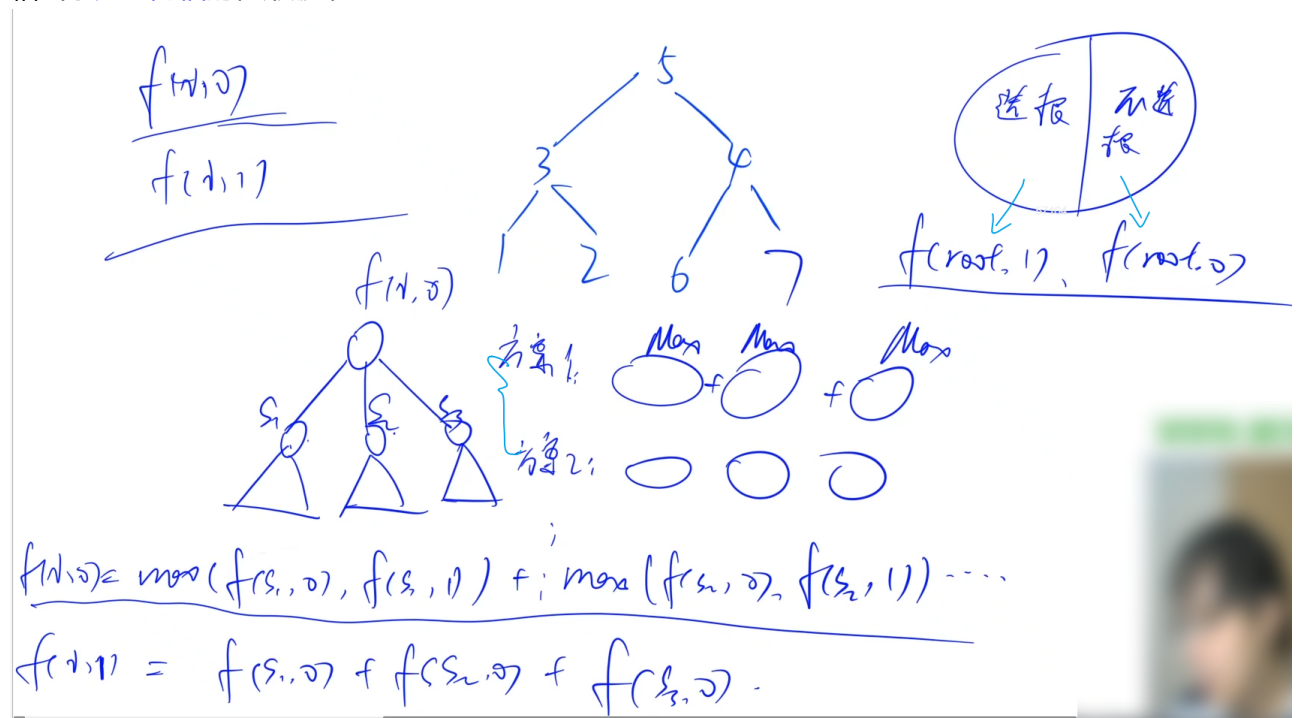
## 285. 没有上司的舞会

### • 题意

Ural 大学有  $N$  名职员，编号为  $1 \sim N$ 。他们的关系就像一棵以校长为根的树，父节点就是子节点的直接上司。每个职员有一个快乐指数，用整数  $H_i$  给出，其中  $1 \leq i \leq N$ 。现在要召开一场周年庆宴会，不过，没有职员愿意和直接上司一起参会。在满足这个条件的前提下，主办方希望邀请一部分职员参会，使得所有参会职员的快乐指数总和最大，求这个最大值。

### • 思路

相当于大盗阿福的树形版本



ps:

- 一般是先递归，再dp，因为一般是从下到上统计答案

### • 代码

...

Author: NEFU AB-IN

```
Date: 2022-03-13 21:36:50
FilePath: \ACM\Acwing\285.py
LastEditTime: 2022-03-15 20:01:27
'''

import sys

sys.setrecursionlimit(int(2e9))

N = 6100
dp = [[0] * 2 for _ in range(N)]
w = [0] * N
g = [[] for _ in range(N)]
deg = [0] * N

def dfs(u):
    dp[u][0] = 0    #dfs先初始化dp数组
    dp[u][1] = w[u]

    for v in g[u]:
        dfs(v)
        dp[u][0] += max(dp[v][0], dp[v][1])
        dp[u][1] += dp[v][0]

n = int(input())
for i in range(1, n + 1):
    w[i] = int(input()) # 别先存dp数组中

for i in range(n - 1):
    v, u = map(int, input().split())
    g[u].append(v)
    deg[v] += 1

for i in range(1, n + 1):
    if deg[i] == 0:
        dfs(i)
        print(max(dp[i][0], dp[i][1]))
        break
```