

양과 늑대

변수

- lamb → 양 숫자
- wolf → 늑대 숫자

입력

- info → 차례로 양/늑대 정보
- edges → 인접 정보

출력

- 정수 결과값 → 양 최대치

Idea

- 조건부 dfs를 하면 될 것 같고 다음좌표를 저장하는 리스트나 무언가가 필요
- 인접 정보를 받아온 edges를 재가공

첫번째는 부모노드, 자식노드 정보 둘다 저장

두번째는 자식노드의 정보만 저장

이 문제는 부모노드를 지나쳐오면 양이고 늑대고 남지 않기 때문에 기억할 필요가 없다

2번째 방법으로 저장하고 갈 수 있는 인덱스를 저장할 다른 무언가가 필요하다.

```
> [[1, 2], [0, 3, 4], [0, 5, 6], [1, 7], [1, 8],  
   [2], [2, 9], [3], [4], [6, 10], [9]]
```

```
[[1, 2], [3, 4], [5, 6], [7], [8], [], [9], [],  
 [], [10], []]
```

- set형으로 갈 수 있는(양 늑대 수 상관없이) 인덱스를 저장 해보자
→ set형은 iteration동안 바뀔수 없다 (처음 암..)

```
RuntimeError: Set changed size during iteration
```

- 배열로 저장하고 수동으로 중복 값 제거하는 루틴이 필요할 것 같다.
→ 했는데 답이 한개는 맞고 한개는 틀림

테스트 1

입력값 > [0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1], [[0, 1],
[1, 2], [1, 4], [0, 8], [8, 7], [9, 10], [9, 1
1], [4, 3], [6, 5], [4, 6], [8, 9]]

기댓값 > 5

실행 결과 > 테스트를 통과하였습니다.

테스트 2

입력값 > [0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0], [[0, 1], [0,
2], [1, 3], [1, 4], [2, 5], [2, 6], [3, 7], [4,
8], [6, 9], [9, 10]]

기댓값 > 5

실행 결과 > 실행한 결괏값 6이 기댓값 5과 다릅니다.

```
import sys
sys.stdin = open("input.txt", "rt")
input = sys.stdin.readline

info = list(map(int, input().split()))
edges = [list(map(int, input().split())) for _ in range(len(info)-1)]

def solution(info, edges):
    global answer, gblInfo, adj, access, visited
    answer = 0
    gblInfo = info
    size = len(info)
    adj = [[] for _ in range(size)]
    visited = [0]*size
    for node1, node2 in edges:
        adj[node1].append(node2)

    dfs(0, 0, 0, [])

    return answer

def dfs(idx, wolf, lamb, access):
    global answer, gblInfo

    if gblInfo[idx] == 0:
```

```
        lamb += 1
        answer = max(answer, lamb)
    elif gblInfo[idx] == 1:
        wolf += 1
    if wolf >= lamb:
        return

    for ad in adj[idx]:
        access.append(ad)
    print(access, lamb, wolf, idx)
    for ac in access:
        if not visited[ac]:
            visited[ac] = 1
            dfs(ac, wolf, lamb, access)
            visited[ac] = 0
        else:
            access.remove(ac)

print(solution(info, edges))
```