

1167번: 트리의 지름

1167번 제출 맞은 사람 슯코딩 풀이 풀이 작성 풀이 요청 재채점/수정 문제 추천

채점 현황 내 소스 강의▼ 질문 검색 질문 작성

트리의 지름

성공 풀이

| 시간 제한 | 메모리 제한 | 제출 | 정답 | 맞은 사람 | 정답 비율 |
|-------|--------|------|-----|-------|---------|
| 2 초 | 128 MB | 2352 | 876 | 675 | 39.153% |

문제

트리의 지름이란, 트리에서 임의의 두 점 사이의 거리 중 가장 긴 것을 말한다. 트리의 지름을 구하는 프로그램을 작성하시오.

입력

트리가 입력으로 주어진다. 먼저 첫 번째 줄에서는 트리의 정점의 개수 V 가 주어지고 ($2 \leq V \leq 100,000$)둘째 줄부터 V 개의 줄에 걸쳐 간선의 정보가 다음과 같이 주어진다. (정점 번호는 1부터 V 까지 매겨져 있다고 생각한다)

먼저 정점 번호가 주어지고, 이어서 연결된 간선의 정보를 의미하는 정수가 두 개씩 주어지는데, 하나는 정점번호, 다른 하나는 그 정점까지의 거리이다. 예를 들어 네번째 줄의 경우 정점 3은 정점 1과 거리가 2인 간선으로 연결되어 있고, 정점 4과는 거리가 3인 간선으로 연결되어 있는 것을 보여준다. 각 줄의 마지막에는 -1이 입력으로 주어진다. 주어지는 거리는 모두 10,000 이하의 자연수이다.

출력

첫째 줄에 트리의 지름을 출력한다.

문제 유형 : 그래프 탐색

접근 방법 : 가중치가 정해져 있는 그래프를 탐색하는 문제로 파악했다. 입력 형식이 뒤에 -1이 붙어 있고, 노드와 가중치 가 쌍으로 전해지기 때문에 입력 받는 것 부터 불편했다.

입력을 받고 난 후 그냥 첫 노드에서 DFS를 돌렸다. 그러니까 첫 노드에서 제일 먼 점이 나왔다.

그래서 아 모든 노드에서 제일 먼점을 구한 후 그 거리 들 중 제일 큰 것을 뽑아야지! 하고 전체 탐색을 전체 탐색 했다. 당연히 시간 초과가 났다.

조금 더 생각해보니까, 어떤 점에서 가장 먼점을 DFS로 탐색한다. 그후 그점에서 제일 먼 점을 탐색하면 그것이 지름 이었다. 좀 만 더 생각해보면 됐는데 아쉬웠다.

```
n = int(input())
cnj = [[] for i in range (n)]
visit= [False for i in range(n)]
sums = []
for i in range (n) :
    line = [int(i) for i in input()[1:-2].split()]
    t = line.pop(0)-1
    for j in range(0,len(line)-1,2):
        cnj[t].append((line[j],line[j+1]))
def DFS (node,s):
    if visit[node] :return
    visit[node] = True
    for i in cnj[node]:
        DFS(i[0]-1,i[1]+s)
    sums.append((s,node))
visit= [False for i in range(n)]
DFS(0,0)
visit= [False for i in range(n)]
start = (max(sums))
DFS(start[1],0)
print(max(sums)[0])
```

튜플이 해싱이 되고 대소비교가 된다!!! 정말 편하다

예제 입력

```
5
1 3 2 -1
2 4 4 -1
3 1 2 4 3 -1
4 2 4 3 3 5 6 -1
5 4 6 -1
```

예제 출력

```
11
```

힌트

출처

- 문제의 오타를 찾은 사람: ababc1005 WeissBlume
- 문제를 만든 사람: author5

알고리즘 분류

- 트리