

문제 설명

꿀벌은 트리 인형 뽑기를 좋아한다.

트리 인형 뽑기란, 1번 정점을 루트로 하는 트리에서 인형을 뽑는 것이다.

규칙은 다음과 같다.

- 만약 어떤 정점 a의 인형을 뽑았다면, 인형은 인형뽑기 기계에서 나가고 정점 a는 비워진다.
- 만약 어떤 정점 a가 비워졌다면, 정점 a의 하위 정점들에 있는 인형들 중 최소 무게를 가진 인형이 정점 a를 채우고, 하위 정점은 비워진다.
- 만약 어떤 정점 a가 비워졌을 때, 하위 정점에 인형이 없다면, 그 정점은 비어진 상태로 있는다.
- 서로 같은 무게의 인형은 존재하지 않는다.
- 트리의 최대 높이는 100을 넘지 않으며, 처음 트리의 모든 정점에 인형이 존재한다.
- 꿀벌은 자신이 뽑고 싶은 인형을 자신이 쓴 리스트 순서대로 뽑아야 한다

그런데 인형 뽑기 기계를 작동시키기 위해 적당한 양의 전력이 필요하다. 꿀벌은 전기를 아끼기 때문에, 뽑을 때 필요한 전력을 정확히 알고 싶다.

어떤 정점 a에서 인형을 뽑을 때, 사용되는 전력은 (1번 정점에서 a번 정점까지의 경로 길이 + 1)이며, 인형이 하위 노드에서 올라올 때는 중간에 작동하는 센서가 놀랍도록 효율적이므로 추가적으로 전력을 사용할 필요는 없다.

인형 뽑기 기계의 구조와 꿀벌이 갖고 싶은 인형의 리스트가 주어질 때, 인형을 뽑을 때 필요한 전력의 총합을 구해보자!

입출력 조건

입출력 조건

#**입력**

첫 번째 줄에 트리의 정점의 갯수 $N(1 \leq N \leq 10^5)$, 쿼리의 갯수 $Q(1 \leq Q \leq min(N, 20000))$ 가 주어진다.

두 번째 줄부터 N번째 줄 까지 $u, v(1 \leq u, v \leq N, u \neq v)$가 주어지며, u번 정점과 v번 정점이 연결되어 있다는 의미이다.

$N+1$번째 줄부터 $2N$번째 줄까지 1번 정점부터 N번 정점까지 순서대로 정점에 있는 인형의 무게 $W(1 \leq W \leq N, W_i \neq W_j)$가 주어지며, $N+i$번째 줄에 i번 정점에 있는 인형의 무게가 주어진다.

$2N+1$번째 줄부터 $2N+Q$번째 줄에 뽑아야 할 인형의 무게 $W(1 \leq W \leq N, W_i \neq W_j)$가 주어지며, $2N+i$번째 줄의 W는 i번째로 뽑아야 할 인형의 무게이다.

#**출력**

첫 번째 줄에 필요한 전력의 총합을 출력한다. int형의 범위를 넘는 테스트케이스는 존재하지 않는다.

🧐

문제 풀이

#**사용해야 하는 알고리즘**

- $PriorityQueue$
- $Tree$
- $Implementation$
- $GraphTraversal-DFS$

#**풀이**

일단 DFS로 트리를 탐색해서 전처리를 하자.

배열 세 개를 채우고, 배열 하나와 변수 하나를 설정하자.

- $used[i]=(i$번 무게의 인형을 뽑았는가?$)$
- $parent[i]=(i$번 노드의 부모 노드$)$
- $A[W]=(W$의 무게를 가진 인형의 위치$)$
- $Route[i]=(i$번 정점과 1번 정점의 거리$)$
- $ans=0$

그리고 각 트리의 정점 a마다 $PriorityQueue$: $T[a]$를 만들고, $T[a]$에 [하위 정점에 있는 인형의 무게, 하위 정점의 번호]를 넣자.

만약 쿼리 W가 들어온다면, $A[W]$가 W의 무게를 가진 인형의 정점의 위치이다.

$p=A[W]$라고 하자. 전력은 쿼리 당 $((p$번 정점과 1번 정점의 거리$)+1)$이 드므로

$ans=ans+Route[p]$이다. 그리고 p번 정점을 비우고 $used[W]=true$로 하자.

p번 정점이 비워지면 하위 노드 중 최소의 무게를 가진 인형을 뽑아야 하는데 우리는 **우선순위 큐**인 $T[p]$에 하위 정점에 있는 인형의 무게를 모두 넣어뒀기 때문에 $T[p]$에서 원소를 뽑아주고 각각 x,y라고 하면(단, $used[x]$가 $true$이면 한 번 더 원소를 빼야 한다.), 무게가 x인 인형이 p번 정점을 채우므로 $A[x]=p$로 업데이트해주고, $T[parent[p]]$ 에 (x,p)를 넣자. 이젠 정점 y가 비었으므로 정점 y를 탐색하러 가자. p에 y를 대입하자.

그런데 가다 보면 어떤 노드 a에 **우선순위 큐** $T[a]$가 비워져 있거나 중복된 값을 제외했더니 비워져 있을 수도 있다. 이때는 하위 노드가 없거나 하위 노드에 인형이 존재하지 않는다는 의미이므로 탐색을 중지한다.

높이가 최대 100이므로 대략 최악의 시간 복잡도는 $O(100 * log(1000) * 20000) < O(10^8)$이지만, N이 쿼리를 처리할 때마다 작아지기 때문에 저 시간복잡도는 잘 나오지 않는다.

참고) 각 쿼리당 최대 전력의 크기는 100이므로 $-2^{32} < 100 * 20000 = 2000000 < 2^{32} - 1$이므로 애초에 조건에 맞는 테스트 케이스 중 int형 범위를 넘는 테스트 케이스는 만들 수 없다.