“트리와 쿼리” 문제 풀이

작성자: 윤교준

부분문제 1

가능한 트리의 형태가 세 가지밖에 없으므로, 조건문으로 쉽게 해결할 수 있다.

부분문제 2

S에 속한 모든 두 정점쌍에 대하여 S 위에서 연결되어 있는지 여부를 판별하자.

두 정점이 S 위에서 연결되어 있는지는 DFS 등을 이용하면 $\mathcal{O}(N)$에 판별할 수 있다.

따라서, 각 쿼리당 $\mathcal{O}\left(NK^2\right)$에 문제를 해결할 수 있다.

부분문제 3

트리에서 S에 속한 정점만 남기고 다른 모든 정점을 제거하면, 여러 개의 트리로 이루어진 포레스트(Forest)가 된다.

이 포레스트의 연결 컴포넌트를 “S 위에서 연결 컴포넌트”라고 하자.

S의 연결 강도는 S 위에서 연결 컴포넌트 각각의 크기를 통하여 계산할 수 있다.

따라서, S에 속하는 정점만을 사용하여 DFS를 수행하면, 각 쿼리당 $\mathcal{O}(N)$에 문제를 해결할 수 있다.

일반적으로, 정점 집합의 부분 집합 $V' \subseteq V$를 방문하는 DFS의 시간 복잡도는 다음과 같다:

$$\sum_{v \in V'} \deg(v)$$

V'에 속하는 각 정점 v마다 $\deg(v)$ 개의 간선을 참조하기 때문이다.

이 문제의 경우, 일반적인 DFS를 수행한다면, 시간 복잡도는 다음과 같다:

$$\sum_{v \in S} \deg(v)$$

이 값은 집합의 크기 $|S| = K$보다 유의미하게 클 수 있음에 유의하라.

예를 들어, 1번 정점에 다른 모든 정점이 붙어 있는 트리 위에서 $S = \{1, 2, \cdots, K\}$를 생각하자. $\deg(1) = N - 1$이므로 K의 값에 무관하게, degree의 합이 $\mathcal{O}(N)$임을 확인할 수 있다.

부분문제 4

Union-find 기법을 응용하자.

입력에서 주어진 트리에서 아무 정점이나 잡아 루트로 만들자.

S에 속하는 각 정점 v마다, v의 부모 정점이 S에 속하는 경우에만 v와 v의 부모 정점을 union하자.

Union된 컴포넌트의 각 크기를 통하여 S의 연결 강도를 계산할 수 있다.

부분문제 3과 다르게, S에 속하는 각 정점 v에 대하여 $\deg(v)$ 개의 간선이 아닌, 단 하나의 간선만을 참조함에 유의하라.

따라서, 각 쿼리당 $\mathcal{O}(K\alpha(N))$에 문제를 해결할 수 있다.

Union-find를 사용하지 않고 각 쿼리마다 정점을 깊이 순으로 정렬한 후, 깊은 정점부터 컴포넌트의 크기를 계산해 나가면서 $\mathcal{O}(N + \sum_i K_i \log K_i)$에 문제를 해결할 수도 있다. 쿼리를 오프라인으로 받을 경우, 계수 정렬을 사용할 수 있어서 위 알고리즘의 복잡도가 $\mathcal{O}(N + \sum_i K_i)$ 로 최적화된다.