

“중등부 3번 / 고등부 2번. 주유소” 문제 풀이

작성자: 김준원

부분문제 1

부분문제의 조건과 함께 문제를 해석하면, 선형으로 연결되어 있는 N 개의 마을 가운데 연속해서 $k + 1$ 개의 마을에 모두 주유소가 없는 경우가 없도록 마을을 고르는 문제가 된다.

($k + 1$ 의 배수)번 마을에 주유소를 하나씩 설치하는 방법을 생각할 수 있다. 그리고 이 방법이 가장 적은 개수의 마을을 골라 주유소를 설치하는 방법이다. 따라서 $\lfloor \frac{N}{k+1} \rfloor$ 를 출력하면 된다.

부분문제 4

앞으로의 풀이에는 그래프 이론의 용어를 사용한다. 즉, KOI 국가는 트리 형태이며, 마을은 정점, 두 마을을 잇고 있는 도로는 간선으로 표현한다.

주유소를 고를 수 있는 가능한 2^N 가지의 배치를 각각 고려한다. 그래프에서 길이가 K 인 모든 경로를 확인하면서, 각 배치가 가능한 배치인지 판별할 수 있다. 트리에서 두 정점을 잇는 길이 k 의 경로는 N^2 개 이하 존재하므로, 시간 복잡도는 $O(2^N N^2 k)$ 이다.

부분문제 2

부분문제 2에서 간선으로 연결되어 있는 두 정점 가운데 하나 이상에는 주유소가 설치되어 있어야 한다. ‘그리디’하게, 즉 어쩔 수 없을 때에만 주유소를 설치하도록 하는 알고리즘을 통해 목표를 이룰 수 있다.

임의의 정점을 루트로 잡은 뒤 DFS를 수행하자. 그러면 간선을 타고 내려가 리프 정점 (더 이상 타고 내려갈 간선이 없는 정점)에 도달한 뒤 다시 올라오는 과정에 전략을 수행할 수 있다.

- 리프 정점에는 주유소를 설치하지 않는다.
- 내부 정점 v 에 대해, v 의 자녀 정점들에 주유소가 하나도 설치되어 있지 않다면 주유소를 설치한다.

$O(N)$ 의 시간 복잡도로 부분문제 2를 풀 수 있다. 이는 아래와 같이 전체 문제에 대한 풀이로 확장할 수 있다.

부분문제 5, 6

앞선 부분문제 2에서의 최적 전략은, ‘자녀 정점에 주유소가 설치되어 있지 않을 때’ 주유소를 설치하는 것이다. 여기서 일반적인 경우의 최적 전략을 추론할 수 있다. 역시 주유소를 ‘그리디’한 방법으로 배치하는 전략이다.

임의의 정점을 루트로 잡은 뒤 DFS를 수행한다. 이제 다음과 같은 조건을 만족하는 정점 v 에 주유소를 설치하면 된다.

- v 의 각 자녀 정점 w 에 대해, w 를 루트로 하는 서브트리에는 주유소가 설치되어 있지 않은 길이가 k 인 경로가 존재하지 않는다.
- v 를 루트로 하는 서브트리에는 주유소가 설치되어 있지 않은 길이가 k 인 경로가 존재한다.

위 전략을 시행하려면, 어떤 정점 v 에 대해 ‘ v 를 루트로 하는 서브트리에서 주유소가 설치되어 있지 않은 가장 긴 경로의 길이’를 계산할 수 있어야 한다. 이를 직접 (비효율적으로) 계산하면 시간 복잡도에 따라 부분문제 5, 6을 풀 수 있다. 이를 빠르게 계산하면 부분문제 7을 풀 수 있다.

부분문제 3, 7

어떤 정점 v 을 루트로 하는 서브트리에서 가장 긴 경로는 어떤 경로일까? 만약 v 의 자녀 정점이 없다면 가장 긴 경로의 길이는 0일 것이다. v 의 자녀 정점이 하나 있다면, v 에서 시작해서 이 자녀 정점을 지나 가장 깊은 리프 정점에서 끝나는 경로가 가장 긴 경로가 될 것이다. v 의 자녀 정점이 두 개 이상이라면, ‘가장 깊은 리프 정점의 깊이’가 가장 깊은 자녀 정점 두 개를 경유하는 경로, 즉 해당 두 리프 정점을 잇는 경로가 될 것이다.

따라서 가장 긴 경로를 찾기 위해서는, 먼저 각 정점에 대해 ‘이 정점 아래에 있는 가장 깊은 리프 정점의 깊이’를 계산하여야 한다. DFS를 수행하면서 해당 정보를 먼저 계산해 두면, 위 문단의 방법에 따라 가장 긴 경로의 길이도 알 수 있다. 정점 v 에 대해 각 자녀 정점에서 가장 긴 경로의 길이가 k 미만인데 v 에서 가장 긴 경로의 길이가 k 이상이 된다면, 정점 v 에 주유소를 설치한다. v 에 주유소를 설치하고 나면, 다른 정점들에 대해 경로를 계산할 때 v 를 제외해야 한다. 주유소가 설치되어 있지 않은 경로만 생각해야 하기 때문이다. 해당 처리도 v 에 대한 DFS 함수가 반환하고 부모 정점으로 올라가는 시점에 수행할 수 있다. 총 시간 복잡도는 $O(N)$ 이 된다.