**1. One Fire Station**

**문제:** 도시의 건물들은 도로로 연결된 Tree 구조를 가지고 있다.

interior node는 도로 교차로를 뜻하고, terminal node는 건물을 나타낸다.

node사이의 거리는 node를 연결하는 경로에 따라 유클리드 거리로 정의된다.

도시에 소방서 하나를 건설하려고 하는데, 목표는 건물과의 최대 거리가 최소가 되어야 한다.

소방서에 가장 적합한 위치를 찾기 위한 알고리즘을 설계하고, 시간과 공간의 복잡성을 분석하라.

**설계:** 가장 먼 거리에 있는 두 건물을 연결하는 교차로들 중 두 건물까지의 거리차가 가장 작은 node에

소방서가 존재한다. 이를 달리 말하면 가장 먼 두 건물의 거리를 지름으로 변환하여 원을 그리면

그 원안에 다른 건물들이 모두 존재하게 되고, 소방서는 원의 반지름에 가장 가까운 node에

존재한다는 말이다. 그 이유는 소방서가 원의 중심에 존재한다면 어떤 건물이라도 소방서와의

최대거리가 원의 반지름과 같거나 작을 수밖에 없고 만약 소방서가 원의 중심에서 벗어난다면

그만큼의 거리가 멀어지는 node와의 거리에 더해져서 최소거리를 무조건 넘어가버리기 때문이다.

그러면 건물과의 최대 거리의 최소화라는 목적에서 벗어나게 된다.

먼저 원의 지름을 구하기 위해 가장 먼 두 건물을 찾는 방법은 임의의 한 node에서 가장 먼

terminal node를 찾고, 그 다음 그 node로부터 가장 먼 terminal node를 찾으면 된다.

어떤 node에서 출발하든 그 node와 가장 먼 node는 반드시 지름을 담당하는 두 node 중 하나가

될 수밖에 없다. 이를 증명하기 위해서는 앞에서 했던 말의 역이 모순이라는 것을 보이면 된다.

즉, 지름이 (a,b)인 트리에서 임의의 node c가 가장 먼 node를 찾았더니 node a, b가 아닌 임의의

node d라는 것이 모순이라는 것을 보이면 된다.

- 증명 ( d(i,j) : i와 j의 거리 )

node d는 d(c,d)를 통해 node c로부터 가장 멀다고 가정했으니 d(d,c)는 d(d,a)와 d(d,b)보다 크다.

(a,b)와 (c,d)가 중간에 t라는 node에서 만난다고 가정하자.

*\*공통으로 만나는 node는 cycle을 만들지 않기 위해서 반드시 하나만 존재해야 한다.*

원, 라인이(가) 표시된 사진

자동 생성된 설명

(d,t)는 a,b,c로 가는 공통 경로로 무시한다면 d(t,c)가 d(t,a)보다 크다는 것을 알 수 있다.

*\* (c,d)가 가장 멀리 있다고 했으니까*

이를 다시 말하면 b에서 a로 가는 거리보다 b에서 c로 가는 거리가 더 길다는 것을 의미하며,

따라서 처음에 트리의 지름은 (a,b)라는 가정은 모순이 된다.

한 노드에서 가장 멀리 있는 노드를 찾기 위해서는 모든 terminal node를 방문해야 하기 때문에

DFS, BFS 둘 다 상관이 없어서 본인은 더 익숙한 BFS를 사용했다.

트리의 지름을 찾았다면 두 노드를 연결하는 경로에 위치해 있는 노드들을 차례대로 방문하면서

두 노드까지의 거리차가 가장 작은 노드에서 멈춘다. 만약 경로에 위치해 있는 노드의 개수가

홀수라면 중심이 되는 노드가 하나지만 짝수라면 두개가 된다.

두 곳 모두 문제의 목적에 위반되지 않는 위치이므로 어느 곳에 소방서를 건설하든 상관없다.

**시간 복잡도**: 노드의 개수가 N, 노드를 연결하는 선의 개수가 E라면 BFS에서 모든 노드를 방문하기 위해

N번의 연산이 발생할 것이고 모든 간선에 대해서 비교를 해야 하기 때문에 E번의 연산이

발생한다. 우리는 두번의 BFS를 해야 하기 때문에 2(N+E)번의 연산이 필요하다.

그리고 두 건물 사이에 있는 노드의 개수는 최대 N-2개까지 있을 수 있기 때문에

소방서의 위치를 계산하기 위해서는 최대 N-2번의 연산이 필요하다.

따라서 본인의 알고리즘의 최종적인 시간복잡도는 T(N) = 3N+E-2 🡪 O(N)이다.

**공간 복잡도**: 노드 N개와 거리를 저장하는 간선 E개가 필요하기 때문에

공간 복잡도는 T(N) = N+E 🡪 O(N)이다.

**2. Two Fire Stations**

**문제:** 하나의 소방서만으로는 도시에 충분하지 않아 두개의 소방서를 건설하고자 한다.

목표는 이전과 동일하게 모든 건물에서 가장 가까운 소방서까지의 최대 거리를 최소화하는 것이다.

두 소방서에 가장 적합한 위치를 찾는 알고리즘을 설계하고, 시간과 공간의 복잡성을 분석하라.

**설계:** 이번 문제를 풀기 위해서는 두개의 소방서의 위치를 트리의 지름을 기준으로 지름의 경로를 따라

좌우로 이동하면서 양 쪽의 최대 거리가 최소가 되도록 해야 한다. 여기서 소방서들이 지름의 경로를

따라 이동해야 하는 이유는 만약 소방서가 지름의 경로를 따르지 않고 더 작은 최대 거리를 찾는다고

하는 순간 최대 거리는 트리의 반지름보다 커지는 모순이 발생하기 때문이다.

따라서 지름의 경로를 따라 분할하는 것이 최적이다.

이제 분할 지점을 계산하여 두개의 소방서를 배치할 위치를 결정한다.

분할 지점을 기준으로 연결을 끊어 좌우의 두 개의 서브트리를 구성한다. 그 다음 두 서브트리의

센터를 찾아 반지름을 구한다. 두 반지름 중 더 큰 값을 저장한다. 이렇게 모든 분할 지점에 대해서

계산된 최대 반지름 중 가장 작은 분할 지점이 최적의 위치이며, 두 소방서는 각 서브트리의 센터에

존재한다.

**시간 복잡도**: 노드의 개수가 N, 노드를 연결하는 선의 개수가 E라면 BFS에서 모든 노드를 방문하기 위해

N번의 연산이 발생할 것이고 모든 간선에 대해서 비교를 해야 하기 때문에 E번의 연산이

발생한다. 우리는 두번의 BFS를 해야 하기 때문에 2(N+E)번의 연산이 필요하다.

지름 경로 상의 노드의 수를 D라고 한다면 전체 노드 수에서 지름 경로상의 노드를

제외한 나머지 노드 수는 N−D이다. 따라서 각 서브트리 대해 방문해야 하는 노드의 개수도

N-D개이며, 서브트리의 간선의 개수를 M이라고 한다면 간선에 대한 비교를 위해

M번의 연산도 발생한다. 또한 모든 분할에 대한 서브트리의 노드의 개수가 2N-2개이기

때문에 분할된 서브트리의 센터를 찾기 위해 2N-2번의 연산이 필요하다.

따라서 알고리즘의 시간 복잡도는 T(N) = 5N+2E-D+M-2 🡪 O(N)이다.

**공간 복잡도**: 노드 N개, 양방향으로 저장할 간선 2E개, (깊이,부모, max\_subtree\_depth, max\_up\_depth)배열

4N크기, 마지막으로 N크기의 큐가 필요하므로 공간복잡도는 T(N)=6N+2E 🡪 O(N)이다.