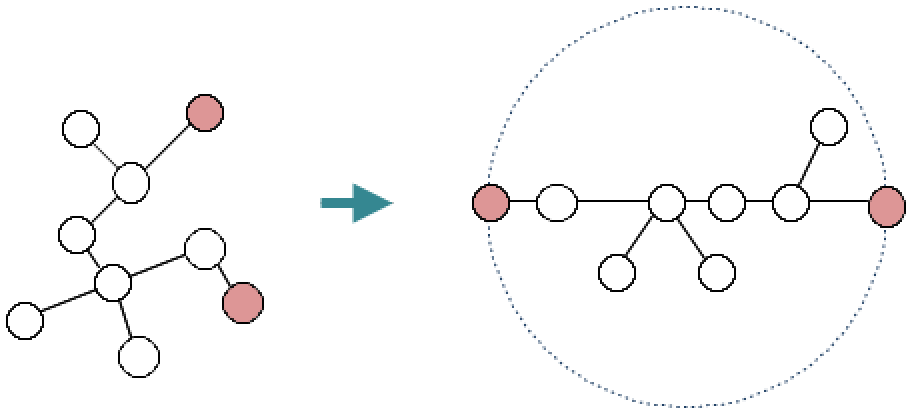
트리의 지름 구하기

<%!마지막 알고리즘 발표%>

트리의 지름



트리의 지름이란?

트리는 사이클이 없는 무방향 그래프이다. 즉 어떤 두 노드를 선택해도 둘 사이에 경로가 항상 하나만 존재하게 된다.

트리에서 어떤 두 노드를 선택해서 양쪽으로 쫙 당길 때, 가장 길게 늘어나는 경우가 있을 것이다. 이럴 때 트리의 모든 노드들은 이 두 노드를 지름의 끝점으로 하는 원안에 들어가게 된다. 이런 두 노드 사이의 경로의 길이를 트리의 지름이라고 한다.

<u>즉, 두 노드를 선택 했을 때 가장 먼 두 정점 사이의 거리를 뜻 한다.</u>

<u>== 트리에 존재하는 모든 경로들 중에서 가장 긴 것의 길이.</u>

트리의 지름

Greedy 방법

1단계 - 트리에서 임의의 정점 a에서 가장 먼 정점 b를 찾는다.

2단계 - b에서 가장 먼 정점 c를 찾는다.

3단계 - b와 c 사이의 거리가 트리의 지름이 된다.

트리의 지름

1단계 - 트리에서 임의의 정점 a에서 가장 먼 정점 b를 찾는다.

2단계 – b에서 가장 먼 정점 c를 찾는다.

3단계 – b와 c 사이의 거리가 트리의 지름이 된다.

증명

어떤 정점 쌍 (v1, v2)가 트리의 지름이라고 하자. 경우의 수는 3가지가 있다.

경우 1) a == v1 || a == v2
경우 2) b == v1 || b == v2
경우 3) v1, v2, a, b가 모두 다른 경우

D(v1,v2) = v1과, v2 사이의 거리.

트리의 지름

1단계 – 트리에서 임의의 정점 a에서 가장 먼 정점 b를 찾는다.

2단계 – b에서 가장 먼 정점 c를 찾는다.

3단계 – b와 c 사이의 거리가 트리의 지름이 된다.

경우 1) a == v1 || a == v2

- v1, v2 중 하나가 a 이므로 ‘D(v1,v2) <= a에서 가장 먼 거리’ 가 성립한다.

  ➔ a에서 다른 정점까지의 거리 <= a에서 가장 먼 거리

- 우리의 정의에 의해 ‘a에서 가장 먼 거리 = D(a, b)’ 이다.

  → D(v1,v2) <= D(a,b)

- b에서 다른 정점 까지의 거리 <= b에서 가장 먼 거리

  → D(a,b) <= D(b,c)

그리고 v1,v2 정의에 의해 ‘D(v1,v2) = 트리의 지름’ 이다.

→ (트리의 지름 ==) D(v1,v2) <= D(a,b) <= D(b,c) 이다.

트리의 지름

1단계 - 트리에서 임의의 정점 a에서 가장 먼 정점 b를 찾는다.

2단계 – b에서 가장 먼 정점 c를 찾는다.

3단계 – b와 c 사이의 거리가 트리의 지름이 된다.

경우 2) b == v1 || b == v2

- v1, v2 중 하나가 b 이므로 'D(v1,v2) <= b에서 가장 먼 거리' 가 성립한다.

  ➔ b에서 다른 정점까지의 거리 <= b에서 가장 먼 거리

- 우리의 정의에 의해 'b에서 가장 먼 거리 = D(b, c)' 이다.

  → D(v1,v2) <= D(b,c)

그리고 v1,v2 정의에 의해 'D(v1,v2) = 트리의 지름' 이다.

→ (트리의 지름 ==) D(v1,v2) <= <u>D(b,c)</u> 이다.

트리의 지름

1단계 – 트리에서 임의의 정점 a에서 가장 먼 정점 b를 찾는다.

2단계 – b에서 가장 먼 정점 c를 찾는다.

3단계 – b와 c 사이의 거리가 트리의 지름이 된다.

경우 3) a, b, v1, v2 가 모두 다른 경우

- a에서 가장 먼 거리 = D(a, b) >= D(a, v2)

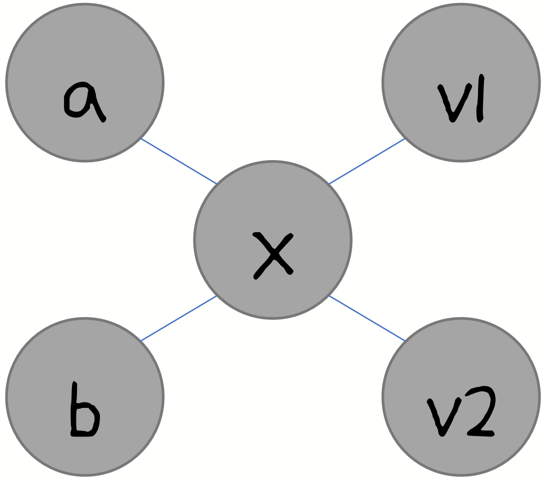
→ D(a, X) + D(X, b) >= D(a, X) + D(X, v2)

→ D(X, b) >= D(X,v2)

→ D(v1, X) + D(X, b) >= D(v1, X) + D(X, v2)

→ D(v1, b) >= D(v1, v2)

→ b에서 v1까지의 거리 >= 트리의 지름

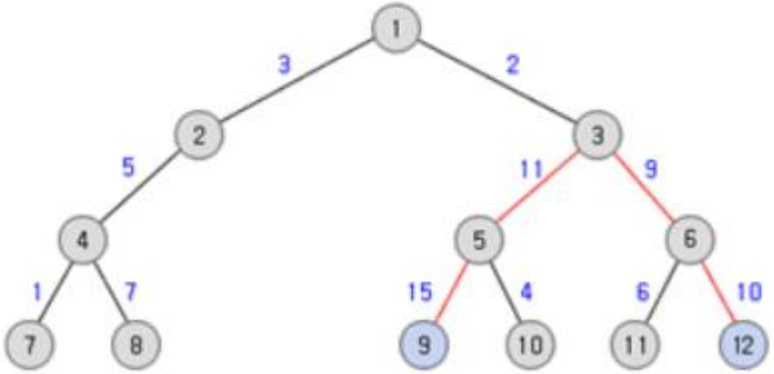


즉, b에서 가장 먼 정점(c)를 찾게 되면 트리의 지름이 된다.

구현

```java
static int dfs(int no) {
    isV[no] = true;
    int lastVertex = no;
    dist[no] = 0;
    for (Node next : graph[no]) {
        if (!isV[next.no]) {
            isV[next.no] = true;
            int vertex = dfs(next.no);
            if (dist[next.no] + next.weight > dist[no]) {
                dist[no] = dist[next.no] + next.weight;
                lastVertex = vertex;
            }
        }
    }
    return lastVertex;
}
```

```java
int b = dfs( no: 1);
Arrays.fill(isV,  val: false);
int c = dfs(b);
System.out.println(dist[b]);
```



G4 1967 트리의 지름

G2 1167 트리의 지름

G1 19581 두 번째 트리의 지름

G2 12912 트리 수정