11064 Diameter

https://acmicpc.net/problem/11064

지름을 어떻게?

트리의 지름을 구하는 방법 중 하나로 DP가 있습니다. 이는 각 서브트리 v마다 "노드의 최대 깊이 $h[v]$"와 "루트를 지나는 경로의 최대 길이 $d[v]$"를 저장하는 방법이죠. v의 자식을 $u_1, \cdots, u_k$, 각 자식으로 가는 간선의 길이를 $c_1, \cdots, c_k$라고 할 때

- $h[v] = max(c_i + h[u_i])$
- $d[v]$는 $c_i + h[u_i]$ 중 가장 큰 두 값의 합 ($k \leq 1$이면 모든 값의 합)

입니다. 그리고 트리의 지름은 모든 $d[v]$ 중 최댓값입니다.

이제 이 풀이를 그대로 이 문제에 옮겨 봅시다. 지름이 D 이하려면, 모든 $d[v]$가 D 이하여야 합니다. 그러므로 각 서브트리마다 "$d[v] \leq D$가 되도록 지불해야 하는 최소 비용 $A[v]$", "최소 비용을 지불했을 때 $h[v]$의 최솟값 $h^-[v]$를 저장합시다.

그게 돼요?

그러면 당연히 "꼭 각 서브트리마다 최소 비용만 지불해야 하는가? 더 지불해서 $h[v]$를 더욱 줄이면 안 되나?" 라는 의문이 들 것입니다. 다행히도 최소 비용만 지불해도 됩니다.

왜냐? 한 서브트리 u_i에서 최소 비용보다 더 지불해서 얻는 이점은 $h[u_i]$가 줄어든다는 것밖에 없습니다. $d[u_i]$가 D 밑으로 계속 줄어드는 건 의미가 없고, $A[u_i]$, $h^-[u_i]$는 하나의 고정된 값이니까요. 하지만 그럴 바에는 그 서브트리를 그대로 놔두고, v와 u_i를 연결하는 간선의 가중치 c_i를 낮추면 됩니다. 그래도 $h[u_i]$가 줄어들 테니까요.

DP식

자식이 없다면, $A[v] = h^-[v] = 0$입니다.

자식이 하나라면, $A[v] = max(0, h^-[u_1] + c_1 - D), h^-[v] = min(h^-[u_1] + c_1, D)$입니다.

자식이 여럿이라면, $h^-[u_1], \cdots, h^-[u_k]$를 모아 내림차순으로 정렬한 배열을 $[h_1, \cdots, h_k]$라고 합시다.. 이 배열의 수들을 잘 줄여서 가장 큰 두 수의 합이 D 이하가 되어야 합니다. 풀어보면 이렇게 됩니다.

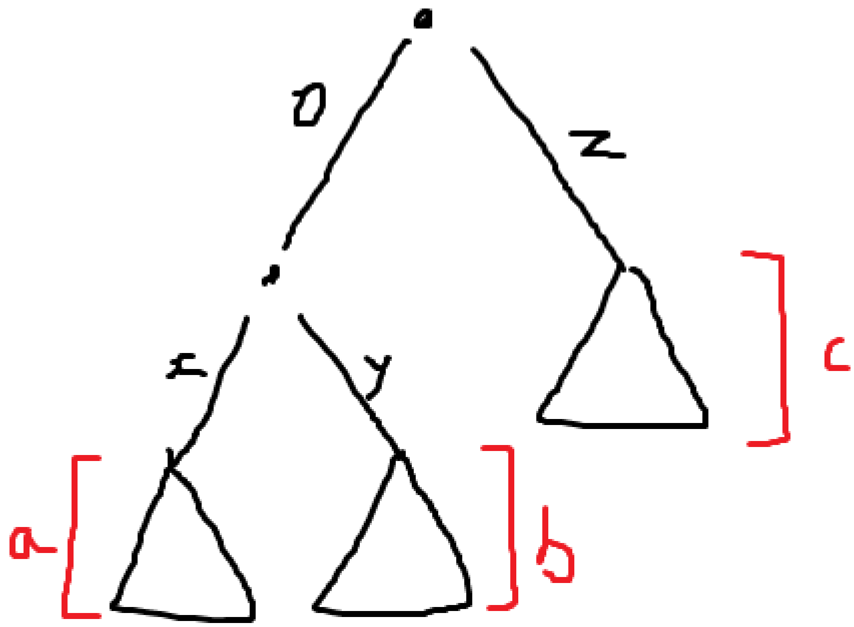
- $2h_2 \leq D$라면, $h_1 + h_2 \leq D$가 될 때까지 h_1만 쭉 내리면 됩니다.

- 아니라면, h_i들 중에서 $\frac{D}{2}$ 이상인 값들을 전부 $\frac{D}{2}$로 내리면 됩니다.

구멍 메우기

사실 여기에는 하나의 허점이 있습니다. c_i가 이미 0이라서 낮출 수 없을 때 문제가 됩니다.

하지만 그래도 DP식은 올바릅니다. 왜일까요?



$0, x, y, z$를 간선의 가중치, a, b, c를 서브트리의 높이라고 합시다. 간선의 가중치가 0이 되었는데도 더 낮춰야 된다는 것은 $max(x+a, y+b)+(z+c) > D$, $max(x+a, y+b) \geq z+c$라는 것입니다. 그러면 $max(x+a, y+b) > \frac{D}{2}$입니다. 그런데 $x+a, y+b$가 모두 $\frac{D}{2}$ 이상일 수는 없습니다. 그러면 $(x+a)+(y+b) > D$가 되어서 서브트리의 모든 지름이 D 이하라는 가정에 모순이기 때문입니다.

따라서 $x+a, y+b$의 값은 같지 않고, 둘 중 큰 쪽을 찾아서 0 대신에 x 또는 y를 대신 낮추면 됩니다. 즉 가중치가 음수로 내려가는 것이 허용되든 안 되든 최소 비용은 달라지지 않습니다.