섞인 트리에서 목표 정점 찾기 (가제)

3초? / 1024MB? / 투스텝

이 문제는 투스텝 문제입니다.

N개의 정점으로 이루어진 트리 T가 있습니다. T의 모든 정점의 차수는 3을 넘지 않습니다. 트리의 각 정점은 흰색 혹은 검은색으로 칠해질 수 있습니다. 처음에 A에게 각 정점에 임의의 순서대로 번호가 붙어 있고, 임의로 색칠된 트리 T가 주어집니다. 그리고 목표 정점 X의 번호가 주어집니다.

A는 T의 정점을 골라 색깔을 바꾸는 작업을 **최대 32번** 할 수 있습니다.

그레이더는 A가 고른 정점의 번호들을 받아 색깔을 바꿉니다. 그리고 각 정점의 번호를 무작위로 새로 부여한 뒤, 간선 리스트의 순서도 무작위로 섞습니다. 그 다음, 그레이더는 B에게 번호가 다르게 부여되고 간선의 순서가 바뀐 트리 T와 채색 정보를 줍니다.

B는 트리 T를 잘 관찰하여 목표 정점 X의 새로 부여된 번호를 맞혀야 합니다. A와 B는 모든 작업동안 소통할 수 없습니다. 단, 모든 작업 이전에 A와 B는 전략을 공유할 수 있습니다.

제약 조건

- $1 \le N \le 1\,000\,000$.
- T의 모든 정점의 차수는 3을 넘지 않는다.

첫 번째 실행

입력

첫 번째 줄에는 첫 번째 실행을 나타내는 문자열 “`first`”가 주어집니다.

두 번째 줄에는 테스트 케이스의 수 C가 주어집니다.

이어서 C개의 블럭이 줄바꿈 문자로 나뉘어 주어집니다. 각 블럭은 다음과 같습니다:

- 첫 번째 줄에는 정점의 개수 N과 목표 정점 X의 번호가 주어집니다.
- 두 번째 줄에는 0과 1로 이루어진 배열 A_i가 주어집니다. $A_i = 0$라면 정점 i는 흰색으로 칠해져 있고, $A_i = 1$일 경우 정점 i는 검은색으로 칠해져 있습니다.
- 이어서 $N - 1$개의 줄에 간선을 나타내는 두 정수 U, V가 공백으로 나뉘어 주어집니다.

출력

C개의 각 i번째 줄에 테스트 케이스 i에 대해 선택한 정점의 나열을 공백으로 구분하여 순서대로 출력합니다. 한 정점의 색을 여러번 바꾸는 것도 가능합니다. 변경 횟수가 32를 넘어가면 `WA` 판정을 받습니다.

두 번째 실행

입력

첫 번째 줄에는 두 번째 실행을 나타내는 문자열 “`second`”가 주어집니다.

두 번째 줄에는 테스트 케이스의 수 C가 주어집니다.

이어서 C개의 블럭이 줄바꿈 문자로 나뉘어 주어집니다. 각 블럭은 다음과 같습니다:

- 첫 번째 줄에는 정점의 개수 N이 주어집니다.
- 두 번째 줄에는 0과 1로 이루어진 배열 A_i가 주어집니다. $A_i = 0$라면 정점 i는 흰색으로 칠해져 있고, $A_i = 1$일 경우 정점 i는 검은색으로 칠해져 있습니다.
- 이어서 $N - 1$개의 줄에 간선을 나타내는 두 정수 U, V가 공백으로 나뉘어 주어집니다.

출력

C개의 각 i번째 줄에 테스트 케이스 i에 대해 알아낸 목표 정점 X의 새로 부여된 번호를 출력합니다.

예제

첫 번째 실행

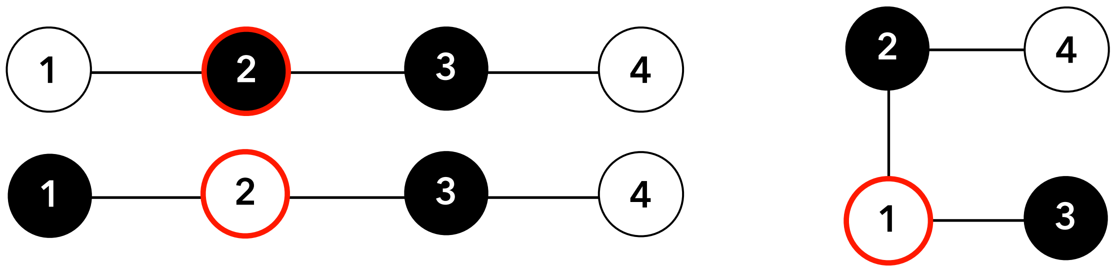
표준 입력(stdin)	표준 출력(stdout)
first 2 4 2 0 1 1 0 1 2 2 3 3 4 5 5 0 1 1 1 1 5 3 2 3 1 2 4 3	1 2 5 1 2 3 3 3

두 번째 실행

표준 입력(stdin)	표준 출력(stdout)
second 5 1 0 0 1 0 5 1 5 3 5 2 4 1 4 1 0 1 0 2 4 2 1 1 3	3 1

테스트 케이스 A 첫 번째 실행: 1 / 두 번째 실행: 2 **설명**

A는 초기 상태에서 1번 정점과 2번 정점의 색깔을 바꿉니다.



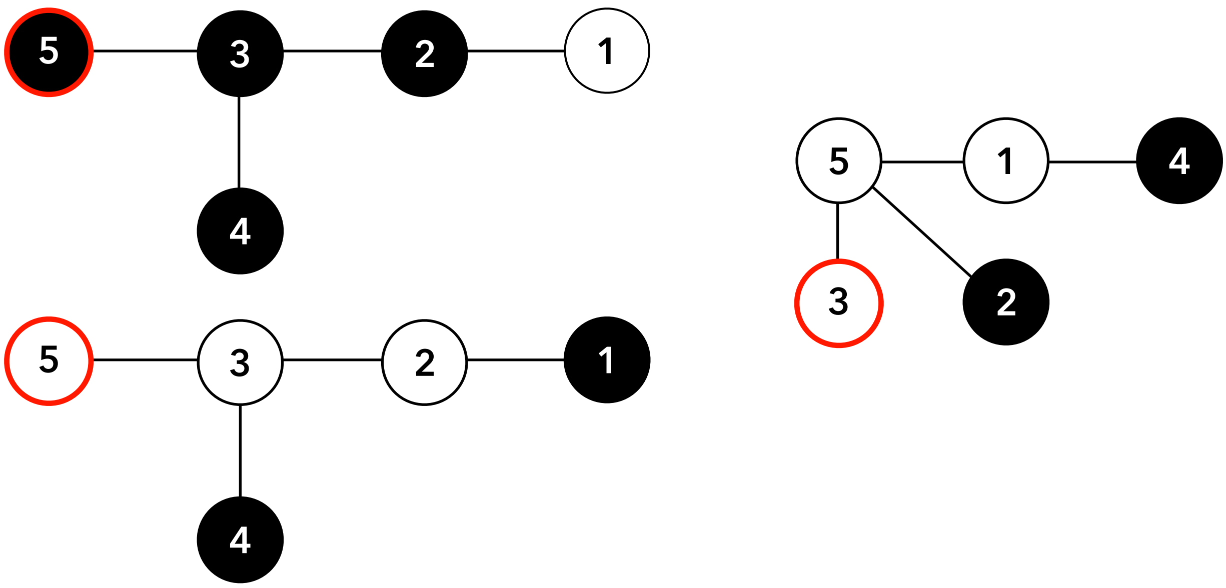
그레이더는 트리의 번호와 간선의 순서를 섞습니다.

초기 상태의 목표 정점 2에 대응되는 새 번호는 1입니다.

따라서 어떤 전략에 따라 B의 입장이 되어 1을 찾아낸 뒤 출력하면 됩니다.

테스트 케이스 B 첫 번째 실행: 2 / 두 번째 실행: 1 **설명**

A는 초기 상태에서 5번, 1번, 2번, 3번 정점의 색깔을 바꿉니다.



그레이더는 트리의 번호와 간선의 순서를 섞습니다.

초기 상태의 목표 정점 5에 대응되는 새 번호는 3입니다.

따라서 어떤 전략에 따라 B의 입장이 되어 3을 찾아낸 뒤 출력하면 됩니다.

섞인 트리에서 목표 정점 찾기 (가제)

태그: centroid, divide and conquer, constructive

트리 크기가 32 이하

따지기 귀찮으니 X만 칠한 상태로 전달합시다.

동형 트리에서 변치 않는 것

트리를 마구 섞어도 센트로이드는 그대로 유지됩니다. 그렇다면, 색칠을 통해 센트로이드와 목표 정점 간의 관계를 잘 설명하여 풀 수 있을 것 같습니다.

센트로이드부터 목표 정점까지

전체 트리 T의 센트로이드를 잡습니다. T부터 X까지 경로는 전부 검은색으로, 경로를 둘러싸는 정점들은 흰색으로 칠해봅시다. 그렇다면 X의 위치를 알아낼 수 있습니다!

하지만 최악의 경우 $O(n)$에 X를 알아내게 됩니다. 따라서 다른 방법이 필요합니다. 센트로이드 분할 과정을 나타낸 센트로이드 트리로 바꿔 생각할 경우, 각 센트로이드를 경로처럼 생각하며 칠해줄 수 있다는 것을 알 수 있습니다. 다행히 차수가 3 이하라는 조건이 있어, 이 경우 $O(\log n)$만큼 색칠을 해두어서 X를 알아낼 수 있습니다.

센트로이드는 최대 2개

하지만 센트로이드는 매번 유일하게 결정되지 않습니다. 단, 그렇게 많지도 않습니다. 최대 2개 존재하고, 그럴 경우 연속하다는 성질을 갖습니다. 그럼 2개의 센트로이드가 발생할 경우에 어떤 센트로이드를 선택해야 할 지 정해야 합니다.

둘의 위치가 정확히 구분되기만 한다면 해결할 수 있는 문제인데, 이는 최초로 잡은 센트로이드를 루트로 한 트리에서 더 높은, 혹은 더 낮은 센트로이드로만 선택하도록 약속해둠으로서 구분할 수 있습니다!

루트가 되는 센트로이드

그렇다면 전체 트리 T의 센트로이드를 유일하게 결정할 방법이 필요합니다. 여기에는 채색 정보를 활용할 수 있습니다. 센트로이드 분할 과정에서 항상 X가 포함된 서브트리로만 이동했기 때문에, 남는 정점들이 존재합니다. 전체 트리 T의 두 연속한 센트로이드의 색을 다르게 하고, 전체 정점의 XOR과 같은 색의 센트로이드를 루트로 특정하면 됩니다. (이 방법이 유일하지는 않겠지만, 이게 제일 간단할 듯 싶습니다) 이러면 최대 2번의 추가적인 채색이 필요합니다.

이때 주의할 점은, A의 입장에서 선택하는 센트로이드가 X와 더 가까운 센트로이드가 되어야 한다는 것입니다. 그렇지 않으면 반대쪽 센트로이드가 X가 될 경우에 제대로 찾을 수 없기 때문입니다.

(이렇게 '어떻게 특정할 것인가'를 두고 세세한 설정을 고려하는 과정이 꽤 어려운 것 같습니다.)

엄밀한 계산

이제 이 방법에 필요한 채색 횟수의 상한을 엄밀하게 계산해봅시다.

최악의 경우 센트로이드 분할 과정에서 트리의 크기가 1이 되기 전까지 센트로이드가 X가 되지 않습니다. 그리고 센트로이드 분할 과정에서 최악의 경우 파고들어가는 서브트리의 크기는 현재 크기가 S라고 할 때, $\lfloor S/2 \rfloor$가 됩니다. 센트로이드는 제외되기 때문이죠.

따라서 경로 속 정점과 경로를 둘러싸는 정점의 개수의 총합은 최대 $3 \times \lfloor \log_2 n \rfloor$가 됩니다.

그리고 앞서 말했듯이 전체 트리 T의 센트로이드를 특정해야 하므로 2번이 더 소요됩니다.

이렇게만 할 경우, $\lfloor \log_2 1\,000\,000 \rfloor = 20$이므로 62번 정도의 색칠이 필요합니다.

하지만 경로 속 정점과 경로를 둘러싼 정점들을 전부 보면서 색깔을 바꿔야 하는 것의 개수를 센 뒤, 그것이 60의 절반인 30를 넘는다면 그 반대들만 색칠을 해줌으로서 동일한 결과를 얻을 수 있습니다.

따라서 총 32번의 색칠으로 X를 찾을 수 있습니다.

정리