

길 찾기 게임

Python3

문제 설명

길 찾기 게임

전무로 승진한 라이언은 기분이 너무 좋아 프렌즈를 이끌고 특별 휴가를 가기로 했다.

내친김에 여행 계획까지 구상하던 라이언은 재미있는 게임을 생각해냈고 역시 전무로 승진할만한 인재라고 스스로에게 감탄했다.

라이언이 구상한(그리고 아마도 라이언만 즐거울만한) 게임은, 카카오 프렌즈를 두 팀으로 나누고, 각 팀이 같은 곳을 다른 순서로 방문하도록 해서 먼저 순회를 마친 팀이 승리하는 것이다.

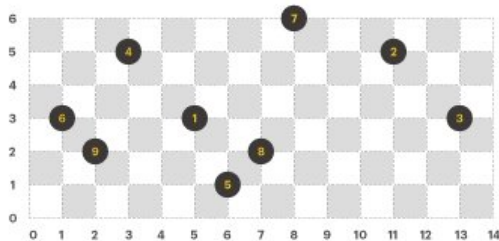
그냥 지도를 주고 게임을 시작하면 재미가 덜해지므로, 라이언은 방문할 곳의 2차원 좌표 값을 구하고 각 장소를 이진트리의 노드가 되도록 구성한 후, 순회 방법을 힌트로 주어 각 팀이 스스로 경로를 찾도록 할 계획이다.

라이언은 아래와 같은 특별한 규칙으로 트리 노드들을 구성한다.

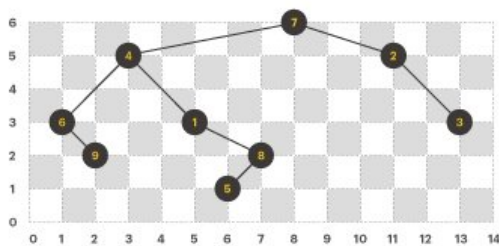
- 트리를 구성하는 모든 노드의 x, y 좌표 값은 정수이다.
- 모든 노드는 서로 다른 x 값을 가진다.
- 같은 레벨(level)에 있는 노드는 같은 y 좌표를 가진다.
- 자식 노드의 y 값은 항상 부모 노드보다 작다.
- 임의의 노드 V 의 왼쪽 서브 트리(left subtree)에 있는 모든 노드의 x 값은 V 의 x 값보다 작다.
- 임의의 노드 V 의 오른쪽 서브 트리(right subtree)에 있는 모든 노드의 x 값은 V 의 x 값보다 크다.

아래 예시를 확인해보자.

라이언의 규칙에 맞게 이진트리의 노드만 좌표 평면에 그리면 다음과 같다. (이진트리의 각 노드에는 1부터 N 까지 순서대로 번호가 붙어있다.)



이제, 노드를 잇는 간선(edge)을 모두 그리면 아래와 같은 모양이 된다.



위 이진트리에서 전위 순회(preorder), 후위 순회(postorder)를 한 결과는 다음과 같고, 이것은 각 팀이 방문해야 할 순서를 의미한다.

- 전위 순회 : 7, 4, 6, 9, 1, 8, 5, 2, 3
- 후위 순회 : 9, 6, 5, 8, 1, 4, 3, 2, 7

다행히 두 팀 모두 머리를 모아 분석한 끝에 라이언의 의도를 간신히 알아차렸다.

그러나 여전히 문제는 남아있다. 노드의 수가 예시처럼 적다면 쉽게 해결할 수 있겠지만, 예상대로 라이언은 그렇게 할 생각이 전혀 없었다.

이제 당신이 나설 때가 되었다.

공경에 빠진 카카오 프렌즈를 위해 이진트리를 구성하는 노드들의 좌표가 담긴 배열 `nodeinfo`가 매개변수로 주어질 때, 노드들로 구성된 이진트리를 전위 순회, 후위 순회한 결과를 2차원 배열에 순서대로 담아 `return` 하도록 `solution` 함수를 완성하자.

제한사항

- `nodeinfo`는 이진트리를 구성하는 각 노드의 좌표가 1번 노드부터 순서대로 들어있는 2차원 배열이다.
 - `nodeinfo`의 길이는 `1` 이상 `10,000` 이하이다.
 - `nodeinfo[i]` 는 `i + 1`번 노드의 좌표이며, `[x축 좌표, y축 좌표]` 순으로 들어있다.
 - 모든 노드의 좌표 값은 `0` 이상 `100,000` 이하인 정수이다.
 - 트리의 깊이가 `1,000` 이하인 경우만 입력으로 주어진다.
 - 모든 노드의 좌표는 문제에 주어진 규칙을 따르며, 잘못된 노드 위치가 주어지는 경우는 없다.

입출력 예

nodeinfo	result
[[5,3],[11,5],[13,3],[3,5],[6,1],[1,3],[8,6],[7,2],[2,2]]	[[7,4,6,9,1,8,5,2,3],[9,6,5,8,1,4,3,2,7]]

입출력 예 설명

입출력 예 #1

문제에 주어진 예시와 같다.

`solution.py`

```
1 class Node:
2     def __init__(self, data, left = None, right = None):
3         self.data = data
4         self.left = left
5         self.right = right
6
7
8 def solution(nodeinfo):
9     for idx, (x, y) in enumerate(nodeinfo, start = 1):
10         idx = Node(idx, )
11
12
```

실행 결과

테스트 1

입력값 > [[5, 3], [11, 5], [13, 3], [3, 5], [6, 1], [1, 3], [8, 6], [7, 2], [2, 2]]
기댓값 > [[7, 4, 6, 9, 1, 8, 5, 2, 3], [9, 6, 5, 8, 1, 4, 3, 2, 7]]
실행 결과 > 실행한 결과값 `null`이(가) 기댓값 `[[7,4,6,9,1,8,5,2,3],[9,6,5,8,1,4,3,2,7]]`와(과) 다릅니다.
출력 > [[6, 1], [7, 2], [2, 2], [5, 3], [13, 3], [1, 3], [11, 5], [3, 5], [8, 6]]
 [[1, 3], [2, 2], [3, 5], [5, 3], [6, 1], [7, 2], [8, 6], [11, 5], [13, 3]]

초기화

코드 실행

제출 후 채점하기