

기둥과 보 설치

Python3

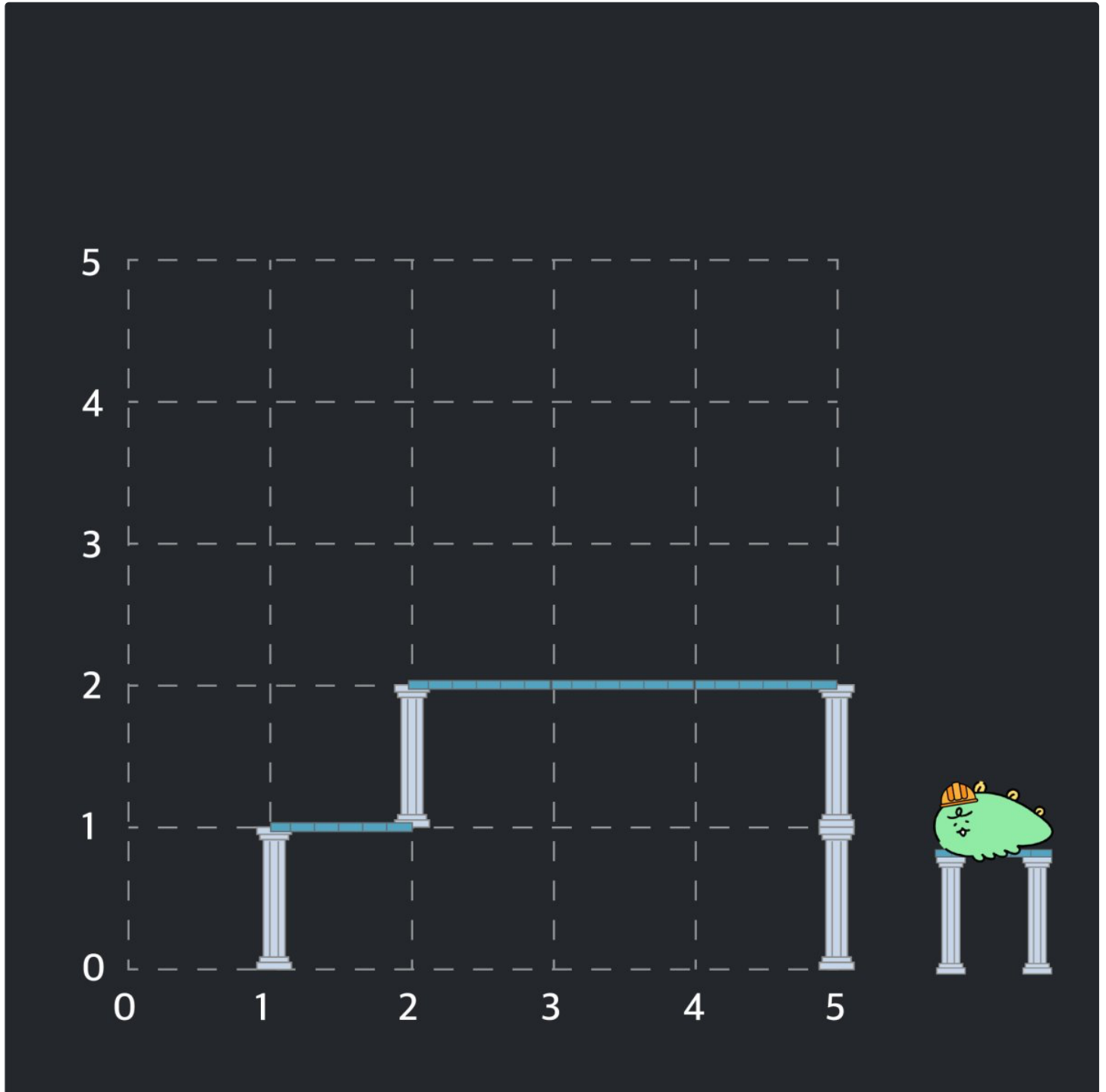
문제 설명

빙하가 깨지면서 스노우타운에 떠나려 온 "조르디"는 인생 2막을 위해 주택 건축사업에 뛰어들기로 결심하였습니다. "조르디"는 **기둥과 보**를 이용하여 벽면 구조물을 자동으로 세우는 로봇을 개발할 계획인데, 그에 앞서 로봇의 동작을 시뮬레이션 할 수 있는 프로그램을 만들고 있습니다. 프로그램은 **2차원 가상 벽면**에 기둥과 보를 이용한 구조물을 설치할 수 있는데, 기둥과 보는 **길이가 1인 선분**으로 표현되며 다음과 같은 규칙을 가지고 있습니다.

- 기둥은 바닥 위에 있거나 보의 한쪽 끝 부분 위에 있거나, 또는 다른 기둥 위에 있어야 합니다.
- 보는 한쪽 끝 부분이 기둥 위에 있거나, 또는 양쪽 끝 부분이 다른 보와 동시에 연결되어 있어야 합니다.

단, 바닥은 벽면의 맨 아래 지면을 말합니다.

2차원 벽면은 $n \times n$ 크기 정사각 격자 형태이며, 각 격자는 1×1 크기입니다. 맨 처음 벽면은 비어있는 상태입니다. 기둥과 보는 격자선의 교차점에 걸치지 않고, 격자 칸의 각 변에 정확히 일치하도록 설치할 수 있습니다. 다음은 기둥과 보를 설치해 구조물을 만든 예시입니다.



예를 들어, 위 그림은 다음 순서에 따라 구조물을 만들었습니다.

1. (1, 0)에서 위쪽으로 기둥을 하나 설치 후, (1, 1)에서 오른쪽으로 보를 하나 만듭니다.
2. (2, 1)에서 위쪽으로 기둥을 하나 설치 후, (2, 2)에서 오른쪽으로 보를 하나 만듭니다.
3. (5, 0)에서 위쪽으로 기둥을 하나 설치 후, (5, 1)에서 위쪽으로 기둥을 하나 더 설치합니다.
4. (4, 2)에서 오른쪽으로 보를 설치 후, (3, 2)에서 오른쪽으로 보를 설치합니다.

만약 (4, 2)에서 오른쪽으로 보를 먼저 설치하지 않고, (3, 2)에서 오른쪽으로 보를 설치하려 한다면 2번 규칙에 맞지 않으므로 설치가 되지 않습니다. 기둥과 보를 삭제하는 기능도 있는데 기둥과 보를 삭제한 후에 남은 기둥과 보들 또한 위 규칙을 만족해야 합니다. 만약, 작업을 수행한 결과가 조건을 만족하지 않는다면 해당 작업은 무시됩니다.

벽면의 크기 n , 기둥과 보를 설치하거나 삭제하는 작업이 순서대로 담긴 2차원 배열 `build_frame`이 매개변수로 주어질 때, 모든 명령어를 수행한 후 구조물의 상태를 `return` 하도록 `solution` 함수를 완성해주세요.

제한사항

- n 은 5 이상 100 이하인 자연수입니다.
- `build_frame`의 세로(행) 길이는 1 이상 1,000 이하입니다.
- `build_frame`의 가로(열) 길이는 4입니다.
- `build_frame`의 원소는 `[x, y, a, b]`형태입니다.
 - x, y 는 기둥, 보를 설치 또는 삭제할 교차점의 좌표이며, [가로 좌표, 세로 좌표] 형태입니다.
 - a 는 설치 또는 삭제할 구조물의 종류를 나타내며, 0은 기둥, 1은 보를 나타냅니다.
 - b 는 구조물을 설치할 지, 혹은 삭제할 지를 나타내며 0은 삭제, 1은 설치를 나타냅니다.
 - 벽면을 벗어나게 기둥, 보를 설치하는 경우는 없습니다.
 - 바닥에 보를 설치 하는 경우는 없습니다.
- 구조물은 교차점 좌표를 기준으로 보는 오른쪽, 기둥은 위쪽 방향으로 설치 또는 삭제합니다.
- 구조물이 겹치도록 설치하는 경우와, 없는 구조물을 삭제하는 경우는 입력으로 주어지지 않습니다.
- 최종 구조물의 상태는 아래 규칙에 맞춰 `return` 해주세요.
 - `return` 하는 배열은 가로(열) 길이가 3인 2차원 배열로, 각 구조물의 좌표를 담고있어야 합니다.
 - `return` 하는 배열의 원소는 `[x, y, a]` 형식입니다.
 - x, y 는 기둥, 보의 교차점 좌표이며, [가로 좌표, 세로 좌표] 형태입니다.
 - 기둥, 보는 교차점 좌표를 기준으로 오른쪽, 또는 위쪽 방향으로 설치되어 있음을 나타냅니다.
 - a 는 구조물의 종류를 나타내며, 0은 기둥, 1은 보를 나타냅니다.
 - `return` 하는 배열은 x 좌표 기준으로 오름차순 정렬하며, x 좌표가 같을 경우 y 좌표 기준으로 오름차순 정렬해주세요.
 - x, y 좌표가 모두 같은 경우 기둥이 보보다 앞에 오면 됩니다.

입출력 예

n	build_frame	result
5	[[1,0,0,1],[1,1,1,1],[2,1,0,1],[2,2,1,1],[5,0,0,1],[5,1,0,1],[4,2,1,1],[3,2,1,1]]	[[1,0,0],[1,1,1],[2,1,0],[2,2,1],[3,2,1],[4,2,1],[5,0,0],[5,1,0]]
5	[[0,0,0,1],[2,0,0,1],[4,0,0,1],[0,1,1,1],[1,1,1,1],[2,1,1,1],[3,1,1,1],[2,0,0,0],[1,1,1,0],[2,2,0,1]]	[[0,0,0],[0,1,1],[1,1,1],[2,1,1],[3,1,1],[4,0,0]]

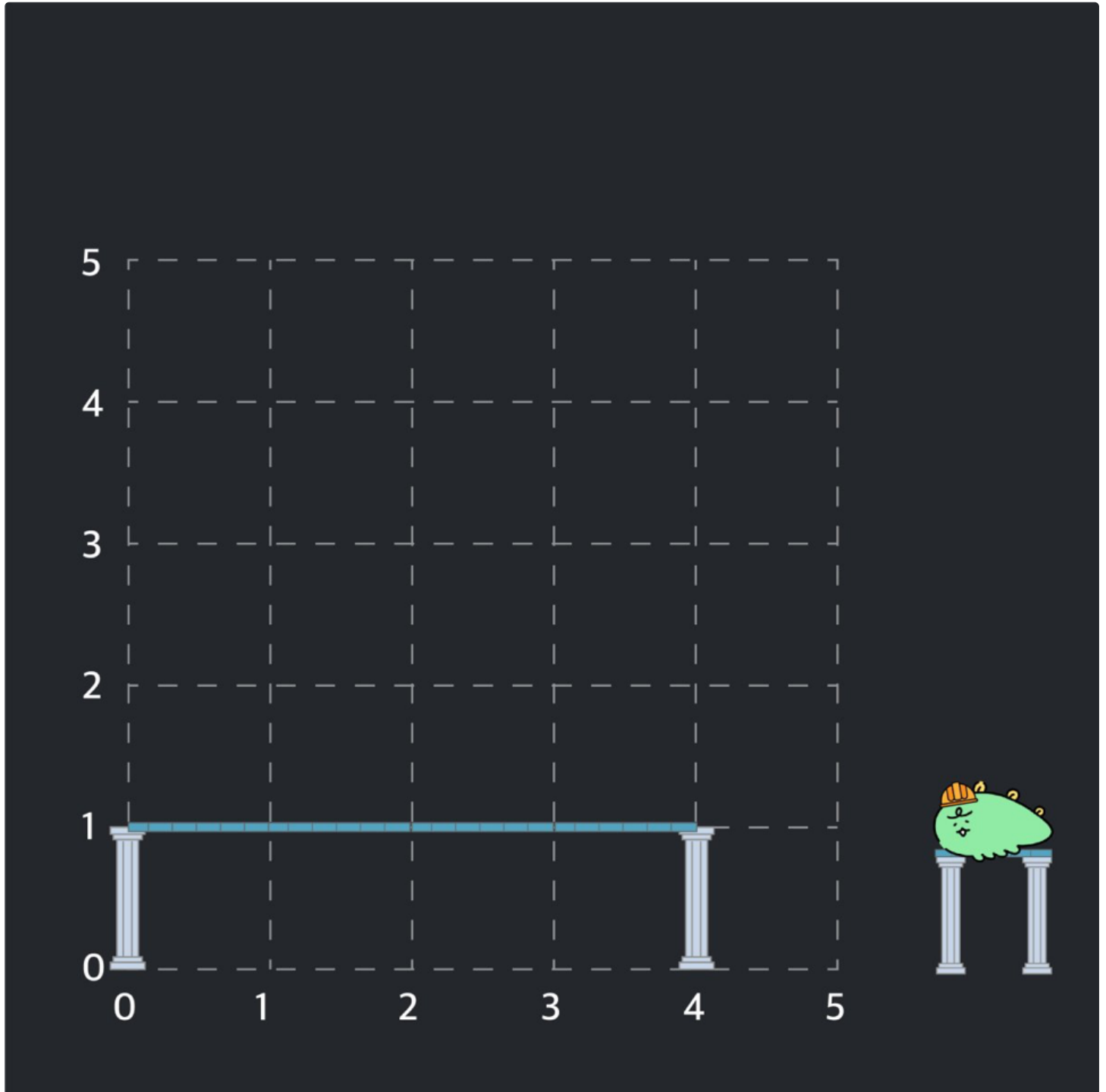
입출력 예에 대한 설명

입출력 예 #1

문제의 예시와 같습니다.

입출력 예 #2

여덟 번째 작업을 수행 후 아래와 같은 구조물 만들어집니다.



아홉 번째 작업의 경우, (1, 1)에서 오른쪽에 있는 보를 삭제하면 (2, 1)에서 오른쪽에 있는 보는 조건을 만족하지 않으므로 무시됩니다.

열 번째 작업의 경우, (2, 2)에서 위쪽 방향으로 기둥을 세울 경우 조건을 만족하지 않으므로 무시됩니다.

혼자 풀기가 막막하다면, 풀이 강의를 들어보세요 [\(클릭\)](#).

solution.py

```

1 def solution(n, build_frame):
2     n = n
3     ## n => n x n
4     ## build_frame(x,y,a,b) => x,y 좌표 a = 0 기둥, a = 1 보, b = 0 삭제, b = 1 설치
5     ## return (x,y,a) => x,y 교차점 좌표, a = 0 기둥, a = 1 보
6
7     table = [[-1]*n for _ in range (n)]
8     for x,y,a,b in build_frame:
9
10        # 기둥
11        # 기둥은 바닥 위에 있거나 보의 한쪽 끝 부분 위에 있거나, 또는 다른 기둥 위에 있어야 합니다.
12        if a == 0:
13            # 설치
14            if b == 1:
15                if x == 0 or (0<y<n and ((table[x][y-1]== 1 and table[x][y+1] == 0) or (table[x
16][y-1] == 0 and table[x][y+1] == 1))) or (x > 0 and table[x-1] == 0)
17                print(x,y,"기둥을 설치합니다.")
18                table[x][y] = 1
19            # 삭제
20            elif b == 0:
21                print(x,y,"기둥을 삭제합니다.")
22
23        # 보
24        # 보는 한쪽 끝 부분이 기둥 위에 있거나, 또는 양쪽 끝 부분이 다른 보와 동시에 연결되어 있어야 합니다.
25        elif a== 1:
26            # 설치
27            if b == 1:
28                if
29                print(x,y,"보를 설치합니다.")
30            # 삭제
31            elif b == 0:
32                print(x,y,"보를 삭제합니다.")
33
34
35     answer = [[]]
36     return answer

```

실행 결과

테스트 1

입력값 > 5, [[1, 0, 0, 1], [1, 1, 1, 1], [2, 1, 0, 1], [2, 2, 1, 1], [5, 0, 0, 1], [5, 1, 0, 1], [4, 2, 1, 1], [3, 2, 1, 1]]

기댓값 > [[1, 0, 0], [1, 1, 1], [2, 1, 0], [2, 2, 1], [3, 2, 1], [4, 2, 1], [5, 0, 0], [5, 1, 0]]

실행 결과 > 실행한 결과값 [[]]이(가) 기댓값 [[1,0,0],[1,1,1],[2,1,0],[2,2,1],[3,2,1],[4,2,1],[5,0,0],[5,1,0]]와(과) 다릅니다.

출력 > [0, 0, 0, 0, 0]

초기화

코드 실행

제출 후 채점하기