[Algorithm] 2019 KAKAO 개발자 겨울 인턴쉽 - 크레인 뽑기(문제 설명 및 문제 풀이) — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북
만든 날짜: 2021-01-28 오후 9:40
URL: https://continuous-development.tistory.com/223

Algorithm

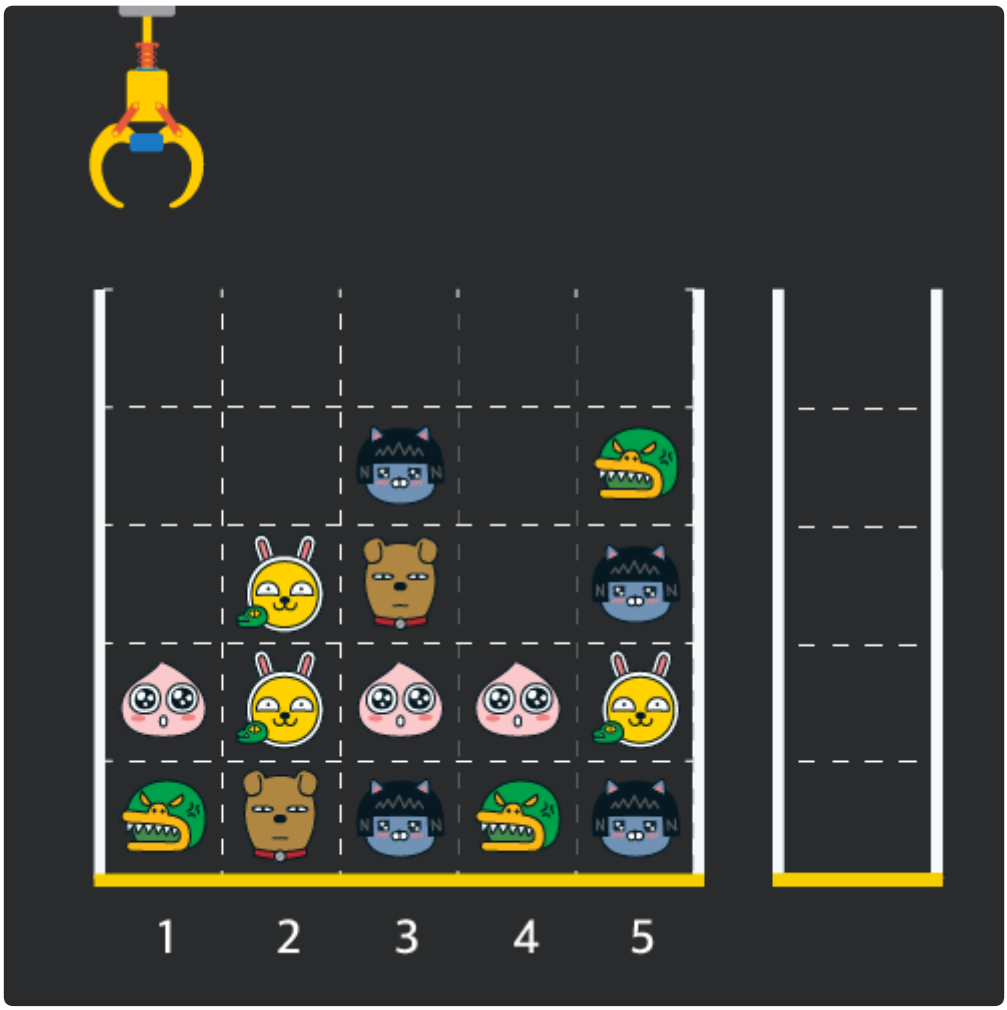
[Algorithm] 2019 KAKAO 개발자 겨울 인턴쉽 - 크레인 뽑기(문제 설명 및 문제 풀이)

2021. 1. 20. 00:23 수정 삭제 공개

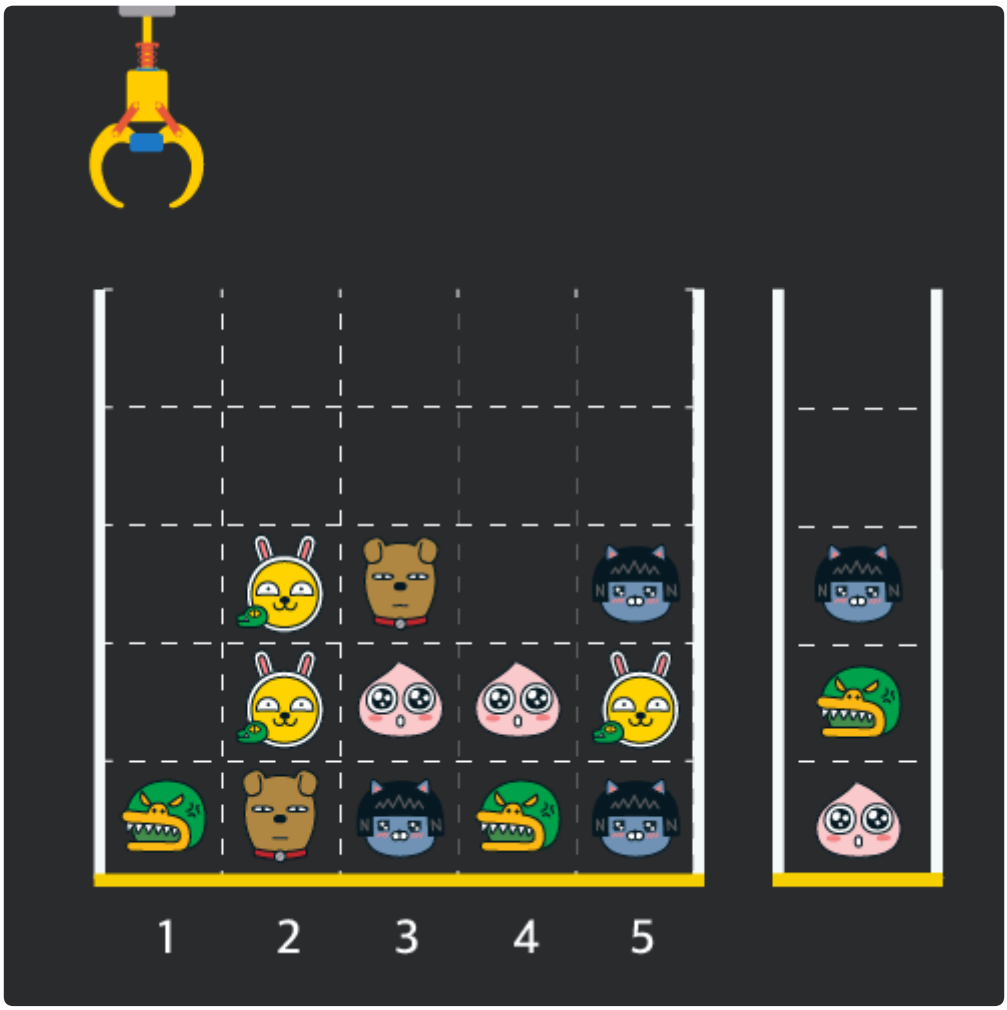
※크레인 뽑기

#문제 설명

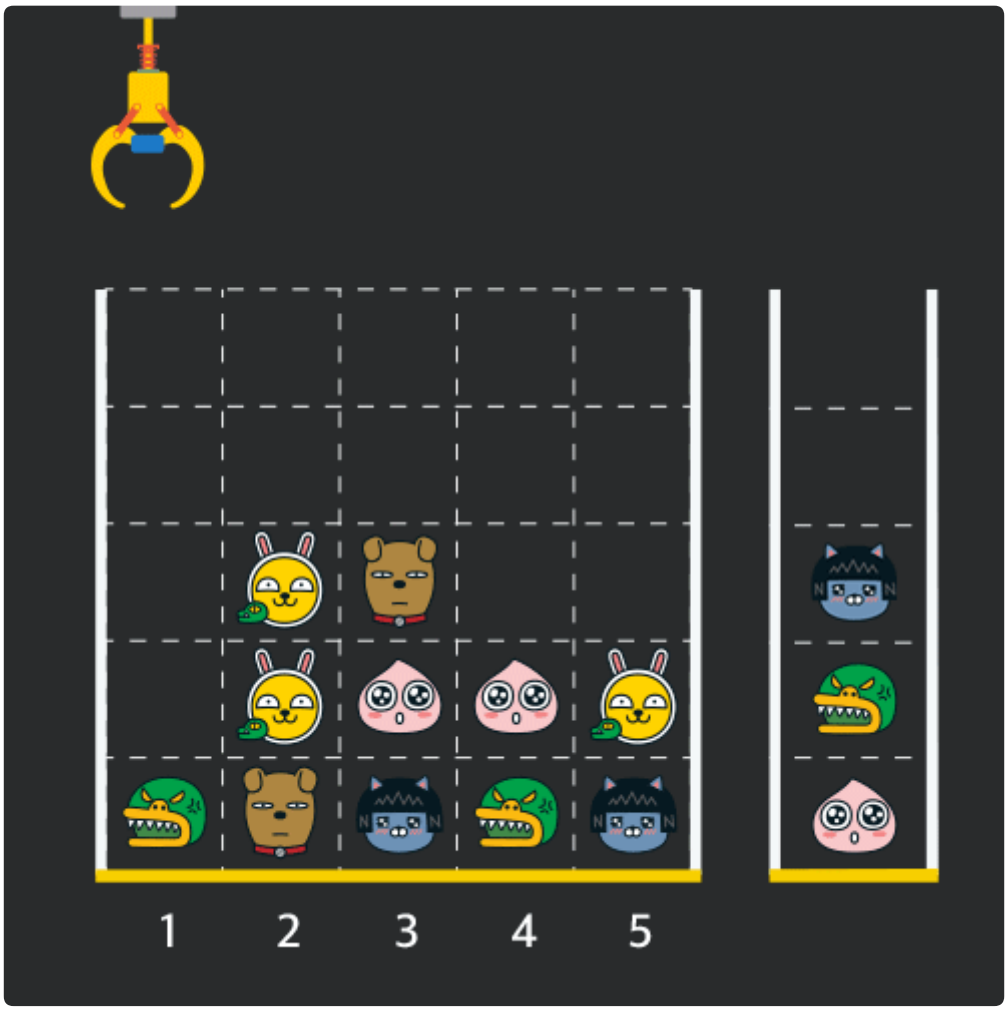
게임개발자인 죠르디는 크레인 인형뽑기 기계를 모바일 게임으로 만들려고 합니다.
죠르디는 게임의 재미를 높이기 위해 화면 구성과 규칙을 다음과 같이 게임 로직에 반영하려고 합니다.



게임 화면은 **1 x 1** 크기의 칸들로 이루어진 **N x N** 크기의 정사각 격자이며 위쪽에는 크레인이 있고 오른쪽에는 바구니가 있습니다. (위 그림은 5 x 5 크기의 예시입니다). 각 격자 칸에는 다양한 인형이 들어 있으며 인형이 없는 칸은 빈칸입니다. 모든 인형은 1 x 1 크기의 격자 한 칸을 차지하며 **격자의 가장 아래 칸부터 차곡차곡 쌓여 있습니다.** 게임 사용자는 크레인을 좌우로 움직여서 멈춘 위치에서 가장 위에 있는 인형을 집어 올릴 수 있습니다. 집어 올린 인형은 바구니에 쌓이게 되는 데, 이때 바구니의 가장 아래 칸부터 인형이 순서대로 쌓이게 됩니다. 다음 그림은 [1번, 5번, 3번] 위치에서 순서대로 인형을 집어 올려 바구니에 담은 모습입니다.



만약 같은 모양의 인형 두 개가 바구니에 연속해서 쌓이게 되면 두 인형은 터뜨려지면서 바구니에서 사라지게 됩니다. 위 상태에서 이어서 [5번] 위치에서 인형을 집어 바구니에 쌓으면 같은 모양 인형 **두 개**가 없어집니다.



크레인 작동 시 인형이 집어지지 않는 경우는 없으나 만약 인형이 없는 곳에서 크레인을 작동시키는 경우에는 아무런 일도 일어나지 않습니다. 또한 바구니는 모든 인형이 들어갈 수 있을 만큼 충분히 크다고 가정합니다. (그림에서는 화면표시 제약으로 5칸만으로 표현하였음)
게임 화면의 격자의 상태가 담긴 2차원 배열 board와 인형을 집기 위해 크레인을 작동시킨 위치가 담긴 배열 moves가 매개변수로 주어질 때, 크레인을 모두 작동시킨 후 터트려져 사라진 인형의 개수를 return 하도록 solution 함수를 완성해주세요.

[제한사항]

- board 배열은 2차원 배열로 크기는 5 x 5 이상 30 x 30 이하입니다.
- board의 각 칸에는 0 이상 100 이하인 정수가 담겨있습니다.
  - 0은 빈 칸을 나타냅니다.
  - 1 ~ 100의 각 숫자는 각기 다른 인형의 모양을 의미하며 같은 숫자는 같은 모양의 인형을 나타냅니다.

- moves 배열의 크기는 1 이상 1,000 이하입니다.
- moves 배열 각 원소들의 값은 1 이상이며 board 배열의 가로 크기 이하인 자연수입니다.

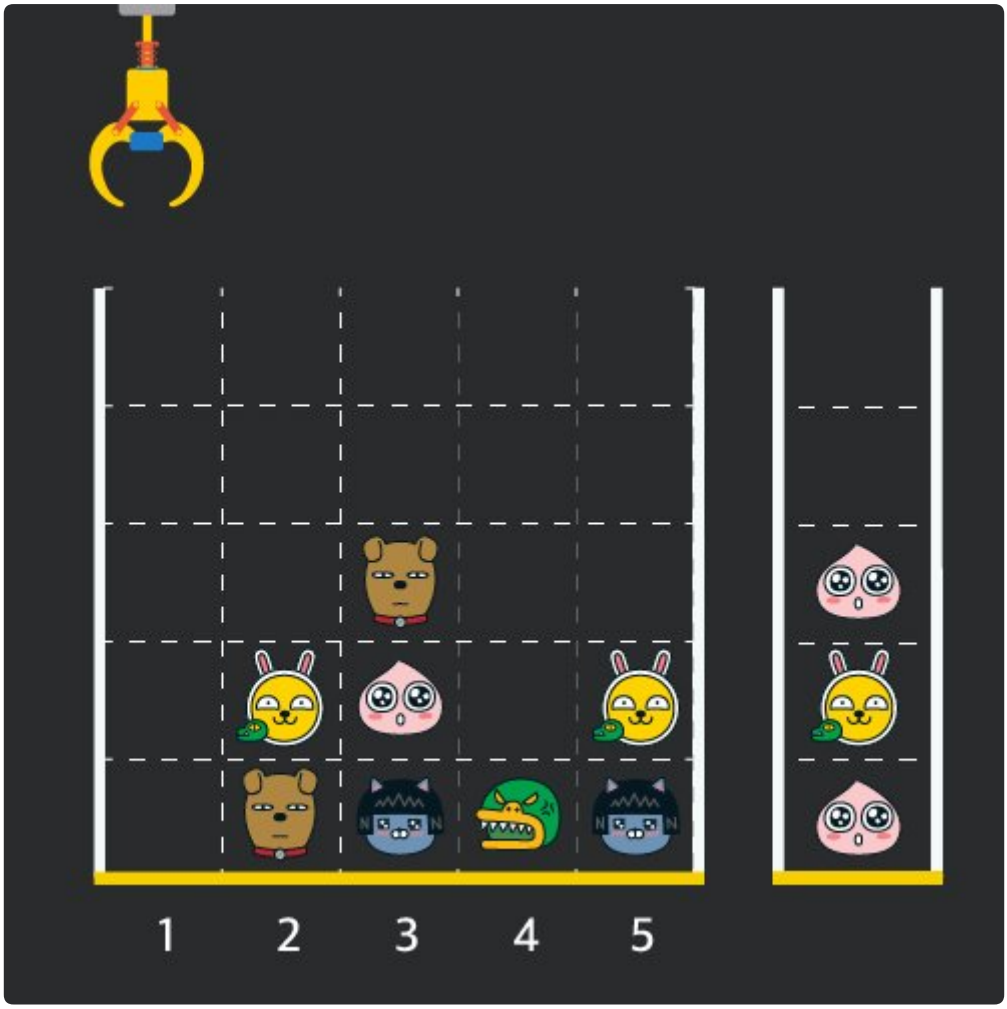
입출력 예

board	moves	result
[[0,0,0,0,0],[0,0,1,0,3],[0,2,5,0,1],[4,2,4,4,2],[3,5,1,3,1]]	[1,5,3,5,1,2,1,4]	4

입출력 예에 대한 설명

입출력 예 #1

인형의 처음 상태는 문제에 주어진 예시와 같습니다. 크레인이 [1, 5, 3, 5, 1, 2, 1, 4] 번 위치에서 차례대로 인형을 집어서 바구니에 옮겨 담은 후, 상태는 아래 그림과 같으며 바구니에 담는 과정에서 터트려져 사라진 인형은 4개 입니다.



문제풀이

문제에서 크레인을 통해 인형을 뽑는다. 이때 move로 해당 값을 가져온다. 그 값은 우리가 받은 행렬로 가지고 오는데 나같은 경우에는 형태에 맞춰서 변환하였다. 해답을 보니 굳이 이럴 필요는 없긴했지만 일단 보기에는 좋긴하다.
moves에 있는 값을 순차적으로 뽑아오고 뽑아온 값은 0 으로 변환을 하였다. 또한 뽑아놓은 바스켓에서는 겹치는 인형들을 삭제하는 방식으로 만들었다.
최종적으로는 삭제된 인형이니 count 했던 값에 *2를 했다.

```
def solution(board, moves):
    modiv_boards=[] # 행렬을 변환하기위해 설정
    for i in range(0,len(board)): # for문을 돌면서 해당 줄에 대한 값을 행렬로 만든다.
        modiv_board=[]
        for j in range(0,len(board)):
            modiv_board.append(board[j][i])
        else:
            modiv_boards.append(modiv_board)
    basket =[]
    count = 0
    for i in range(0,len(moves)): # for문을 통해 moves 만큼 돌게 구현한다.
        for j in range(0,len(modiv_boards)):   # 여기서는 행렬의 길이만큼 돌게한다.
            if modiv_boards[moves[i]-1][j] == 0 : # 이때 행렬을 선택하는데 있어서 moves의 값에 1을 뺀값.
                continue                    # 값이 0 이면 이 부분을 타게 되고
            else:
                number = modiv_boards[moves[i]-1][j] #  0 이 아닌 값이 나오면 그 값을 선택하고 0으로 바꿔
                modiv_boards[moves[i]-1][j] = 0
                basket.append(number)
                if len(basket) >= 2:          # 마지막으로는 길이가 2 이상일때 마지막 인형과 그 마지막에서 2
                    if basket[-1] == basket[-2]: #pop을 통해 값을 빼고 count로 값을 더했다.
                        basket.pop()
                        basket.pop()
                        count += 1
                break
    answer = count*2 # 최종에는 *2를 해서 삭제된 인형의 개수를 구했다.
    print(answer)
    return answer

moves = [1,5,3,5,1,2,1,4]
board=[[0,0,0,0,0],[0,0,1,0,3],[0,2,5,0,1],[4,2,4,4,2],[3,5,1,3,1]]
solution(board, moves)
```

'**Algorithm**' 카테고리의 다른 글

[Algorithm] 2018 KAKAO BLIND RECRUITMENT - 비밀 지도(문제 설명 및 문제...

[Algorithm] 2019 KAKAO 개발자 겨울 인턴쉽 - 크레인 뽑기(문제 설명 및 문...

[Algorithm] 2019 KAKAO BLIND RECRUITMENT - 실패율 (문제 설명 및 문제 ...

[Algorithm] 40강 : 구간 합(Interval) 빠르게 구하기

[Algorithm] 39강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현

[Algorithm] 38강 : 에라토스테네스의 체 알고리즘의 정의와 구현

개발자 인턴쉽 카카오 알고리즘 카카오 인턴쉽 카카오 코딩테스트

나아무늘보

혼자 끄적끄적하는 블로그 입니다.