HA preparation

보드게임

문제

N \* N의 크기를 가진 보드판 위에서 게임을 하려고 합니다. 게임의 룰은 다음과 같습니다.

1. 좌표 왼쪽 상단(0, 0)에 말을 놓는다.
2. 말은 상, 하, 좌, 우로 이동할 수 있고, 플레이어가 조작할 수 있다.
3. 조작의 기회는 딱 한 번 주어진다.
4. 조작할 때 U, D, L, R은 각각 상, 하, 좌, 우를 의미하며 한 줄에 띄어쓰기 없이 써야 한다.
   1. 예시: UDDLLRRDRR, RRRRR
5. 한 번 움직일 때마다 한 칸씩 움직이게 되며, 그 칸 안의 요소인 숫자를 획득할 수 있다.
6. 방문한 곳을 또 방문해도 숫자를 획득할 수 있다.
7. 보드 밖을 나간 말은 OUT 처리가 된다.
8. 칸 안의 숫자는 0 또는 1이다.
   1. 단, 좌표 왼쪽 상단(0, 0)은 항상 0이다.
9. 획득한 숫자를 합산하여 숫자가 제일 큰 사람이 이기게 된다.

보드판이 담긴 board와 조작하려고 하는 문자열 operation이 주어질 때, 말이 해당 칸을 지나가면서 획득한 숫자의 합을 구하는 함수를 작성하세요.

입력

### **인자 1: board**

* number 타입의 2차원 배열
* 2 <= board.length <= 1,000
* 예: [ [0, 0, 1], [1, 0, 1], [1, 1, 1] ]

### **인자 2: operation**

string 타입의 대문자 영어가 쓰여진 문자열

1 <= operation.length <= board.length \* 2

* U, L, D, R 이외의 문자열은 없습니다.

출력

* Number 타입을 반환해야 합니다.
  + board와 operation이 입력값의 예시 ([ [0, 0, 1], [1, 0, 1], [1, 1, 1] ], DDR)일 때, (0, 0) -> (1, 0) -> (2, 0) -> (2, 1) 순서로 이동하게 되고, 각 0, 1, 1, 1을 얻어 3을 반환합니다.

주의사항

* 만약, 말이 보드 밖으로 나갔다면 즉시 OUT 을 반환합니다.

답

function boardGame(board, operation) {

// TODO: 여기에 코드를 작성하세요.

let x = 0, y = 0, point = 0;

for(let i of operation) {

if(i === 'U') {

x -= 1;

} else if(i === 'D') {

x += 1;

} else if(i === 'L') {

y -= 1;

} else if(i === 'R') {

y += 1

}

if(x < 0 || x >= board.length || y < 0 || y >= board[0].length) {

return 'OUT';

} else {

point += board[x][y];

}

}

return point;

};

GCD / LCM 유클리드 호제법 : GCD

function gcd(a, b) {

while (b !== 0) {

let r = a % b;

a = b;

b = r;

}

return a;

}

function gcd(a, b) {

if(b === 0) {

return a;

}

else return gcd(b, a%b);

}

function lcm(a, b) {

return a \* (b / gcd(a, b));

}

순열 nPr

순서를 부여하여 조합할 수 있는 모든 경우의 수

한 번씩은 전부 들어가야 하기 때문에 순회는 필수 반복문, 재귀로 할 수 있다.