1706. Where Will the Ball Fall

Tags	medium
Property	@October 11, 2022
URL	

Question

原文：

You have a 2-D `grid` of size `m x n` representing a box, and you have `n` balls. The box is open on the top and bottom sides.

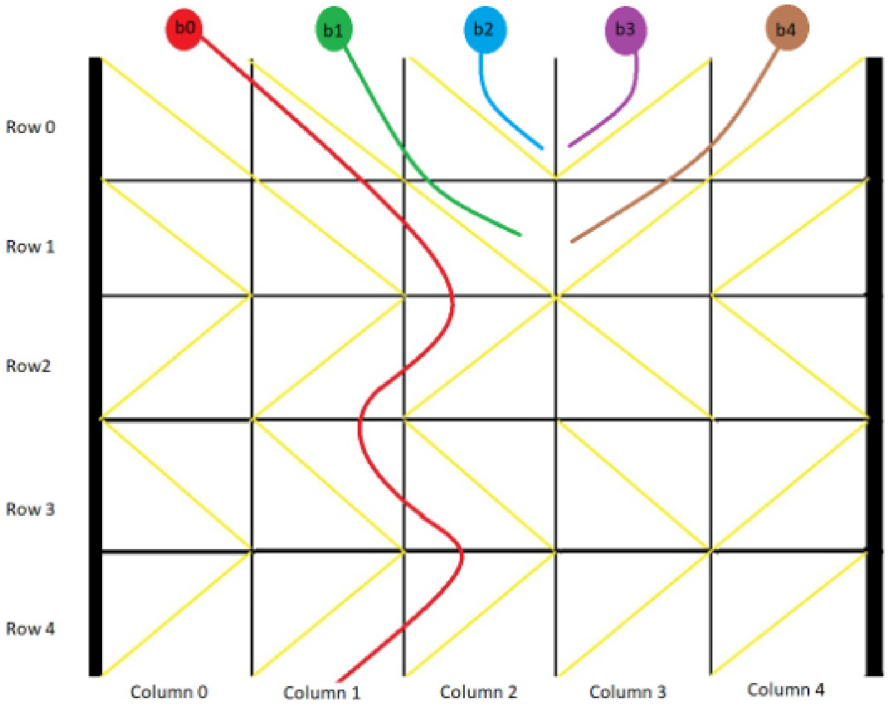
Each cell in the box has a diagonal board spanning two corners of the cell that can redirect a ball to the right or to the left.

- A board that redirects the ball to the right spans the top-left corner to the bottom-right corner and is represented in the grid as `1`.
- A board that redirects the ball to the left spans the top-right corner to the bottom-left corner and is represented in the grid as `1`.

We drop one ball at the top of each column of the box. Each ball can get stuck in the box or fall out of the bottom. A ball gets stuck if it hits a "V" shaped pattern between two boards or if a board redirects the ball into either wall of the box.

Return *an array* `answer` *of size* `n` *where* `answer[i]` *is the column that the ball falls out of at the bottom after dropping the ball from the* `i` `th` *column at the top, or* `-1` *if the ball gets stuck in the box.*

Example 1:



```
Input: grid = [[1,1,1,-1,-1],[1,1,1,-1,-1],[-1,-1,-1,1,1],[1,1,1,1,-1],
[-1,-1,-1,-1,-1]]
Output: [1,-1,-1,-1,-1]
Explanation: This example is shown in the photo.
Ball b0 is dropped at column 0 and falls out of the box at column 1.
Ball b1 is dropped at column 1 and will get stuck in the box between column 2 and 3 and
row 1.
Ball b2 is dropped at column 2 and will get stuck on the box between column 2 and 3 and
row 0.
Ball b3 is dropped at column 3 and will get stuck on the box between column 2 and 3 and
row 0.
Ball b4 is dropped at column 4 and will get stuck on the box between column 2 and 3 and
row 1.
```

我的理解：

給定一個二維vector，這個vector類似於墜落迷宮，並且如果在同列並且特定的相鄰值相同，例如[0][0]等於1那它右邊的值[0][1]如果也等於1它們就可以形成一個往右下的過道，假如[0][1]等於-1且[0][0]等於-1，則可以形成一個向左下過道，如果球可以順著過道而下倒掉出vector就記錄掉出的位置，如上圖第0顆球在第一個位置掉出，會卡在vector中則紀錄為-1

翻譯：

你有一个尺寸为`m x n`的二维`网格`，代表一个盒子，你有`n`个球。盒子的顶部和底部都是开放的。

盒子里的每个单元格都有一个对角板，横跨单元格的两个角，可以将球转到右边或左边。

- 将球转到右边的棋盘横跨左上角到右下角，在网格中表示为 "1"。
- 将球转向左边的棋盘横跨右上角至左下角，在网格中表示为`1'。

我们在盒子的每一列的顶部丢一个球。每个球都可以卡在盒子里或者从底部掉出来。如果球碰到两块木板之间的 "V "形图案，或者一块木板将球重定向到盒子的任何一面墙上，球就会被卡住。

返回一个*数组* `答案` *大小* `n` *其中* `答案[i]` *是球从顶部的* `第*列掉下来后，在底部掉出来的那一列，如果球被卡在盒子里，则`-1`。

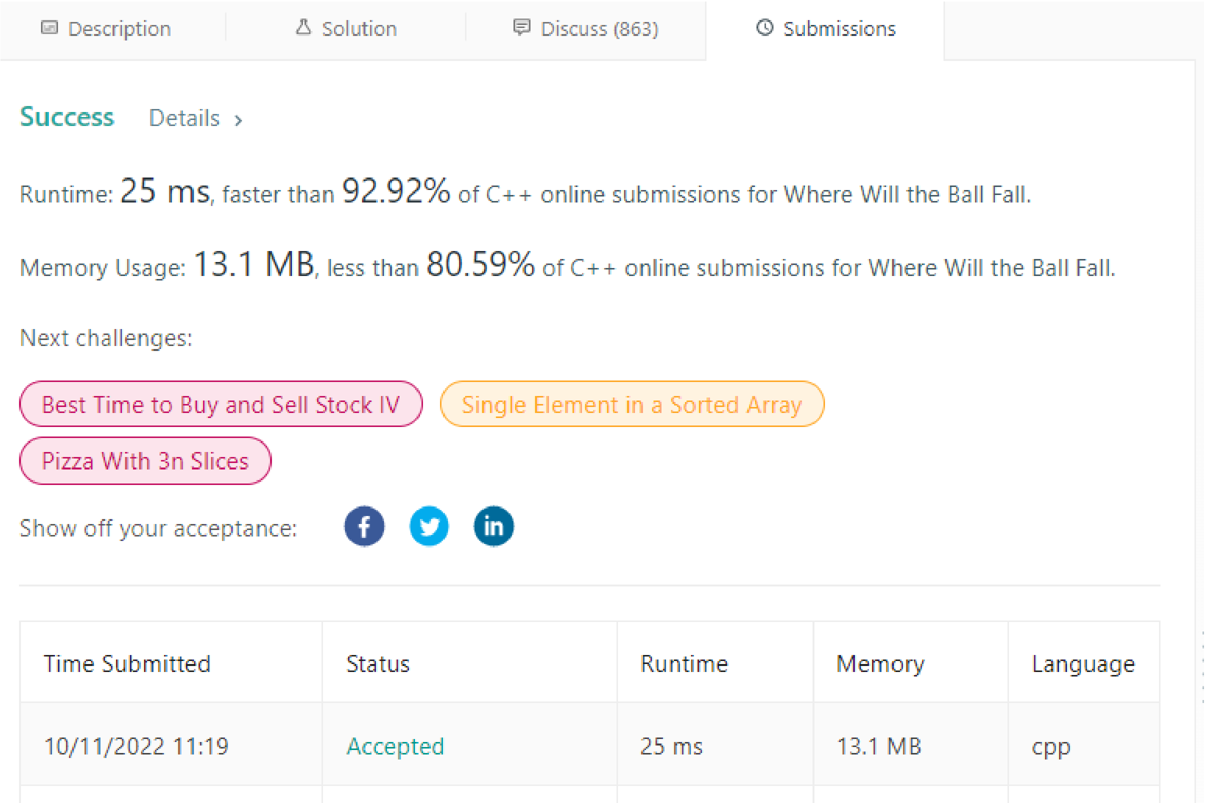
自評翻譯正確性：80(一開始以為是通過就是1，後來發現通過是回復位置

- Word Memory：
    - diagonal 對角

Code

```
class Solution {
public:
    vector<int> findBall(vector<vector<int>>& grid) {
        //i count
        //pass record ball pass or not
        //floor record ball in which floor 0~grid.size()
        //site record ball site in floor 0~grid[0].size()
        int i,pass=1,floor=0,site=0;
        vector<int>store_pass;
        for(i=0;i<grid[0].size();i++){//in order put ball into vector
            site=i;
            pass=1;
            floor=0;
            while(pass==1&&floor<grid.size()){//while pass or ball not yet fall out vector
                //check next floor same site
                //if value == 1 check right side value if value == 1 too
                //mean ball can pass ball go to next floor and move to the rigth side 1 site
                //so floor ++ and side ++
                if(site+1<grid[0].size()&&grid[floor][site]==1){
                    if(grid[floor][site+1]==1){
                        site++;
                        floor++;
                    }
                    //if cannot pass mean ball stock in vector pass change to 0
                    else{
                        pass=0;
                    }
                }
                //check next floor same site
                //if value == -1 check left side value if value == -1 too
                //mean ball can pass ball go to next floor and move to the left side 1 site
                //so floor ++ and side --
                else if(site-1>=0&&grid[floor][site]==-1){
                    if(grid[floor][site-1]==-1){
                        site--;
                        floor++;
                    }
                    else{
                        pass=0;
                    }
                }
                // process edge value
                else{
                    pass=0;
                }
            }
            //if ball pass record where is ball fall out
            if(pass==1){
                store_pass.push_back(site);
            }
            //if ball stock record -1
            else if(pass==0){
                store_pass.push_back(-1);
            }
        }
        return store_pass;
    }
};
```

> 思路：主要就是看球當時腳下的那個數值，假如球當前位置的正下方那格數值是1，代表求知後要向右下移動，那腳下那格的右邊那個數值也要為1才可以往右下，如果是-1的話會變成一個對角將球卡住，反之也是一樣，如果腳下是-1，腳下的左邊也要是-1才可以往左邊掉，還要考慮往右下或左下滾動會不會直接滾到邊界，那也是會卡住，如果你在第0個位置還想往左下就會卡住，所以if有額外寫判定球是不在邊緣，最後如果球可以順利掉出就把site記錄下來，不能的話就紀錄-1，最後輸出。


Description
Solution
Discuss (863)
Submissions

Success Details ›

Runtime: 25 ms, faster than 92.92% of C++ online submissions for Where Will the Ball Fall.

Memory Usage: 13.1 MB, less than 80.59% of C++ online submissions for Where Will the Ball Fall.

Next challenges:

Best Time to Buy and Sell Stock IV
Single Element in a Sorted Array
Pizza With 3n Slices
Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
10/11/2022 11:19	Accepted	25 ms	13.1 MB	cpp

優良code參考

```
class Solution {
public:
    vector<int> findBall(vector<vector<int>>& grid) {
        vector<int> result(grid[0].size(), 0);
        for (int i = 0; i < grid[0].size(); i++) {
            result[i] = findBallDropColumn(0, i, grid);
        }
        return result;
    }

    int findBallDropColumn(int row, int col, vector<vector<int>>& grid) {
        // base case; ball reached the last row
        if (row == grid.size()) return col;
        int nextColumn = col + grid[row][col];
        if (nextColumn < 0 || nextColumn > grid[0].size() - 1 ||
            grid[row][col] != grid[row][nextColumn]) {
            return -1;
        }
        return findBallDropColumn(row + 1, nextColumn, grid);
    }
};
```

> 思路：