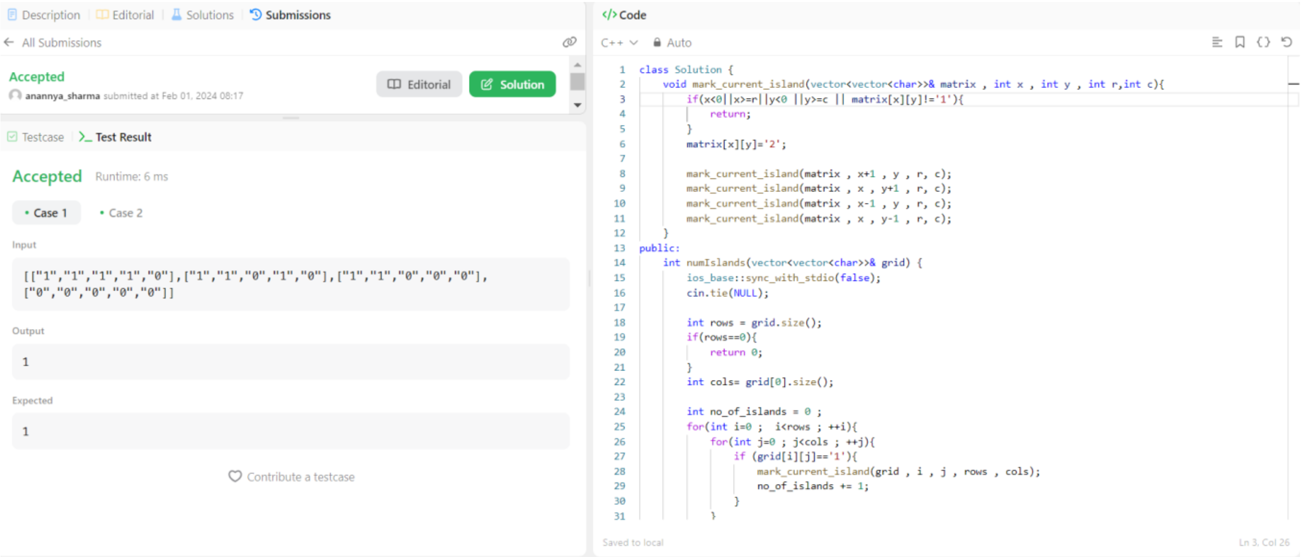
QUES 1



Accepted
anannya_sharma submitted at Feb 01, 2024 08:17

Accepted Runtime: 6 ms

• Case 1 • Case 2

Input

```
[["1","1","1","1","0"],["1","1","0","1","0"],["1","1","0","0","0"],
["0","0","0","0","0"]]
```

Output

```
1
```

Expected

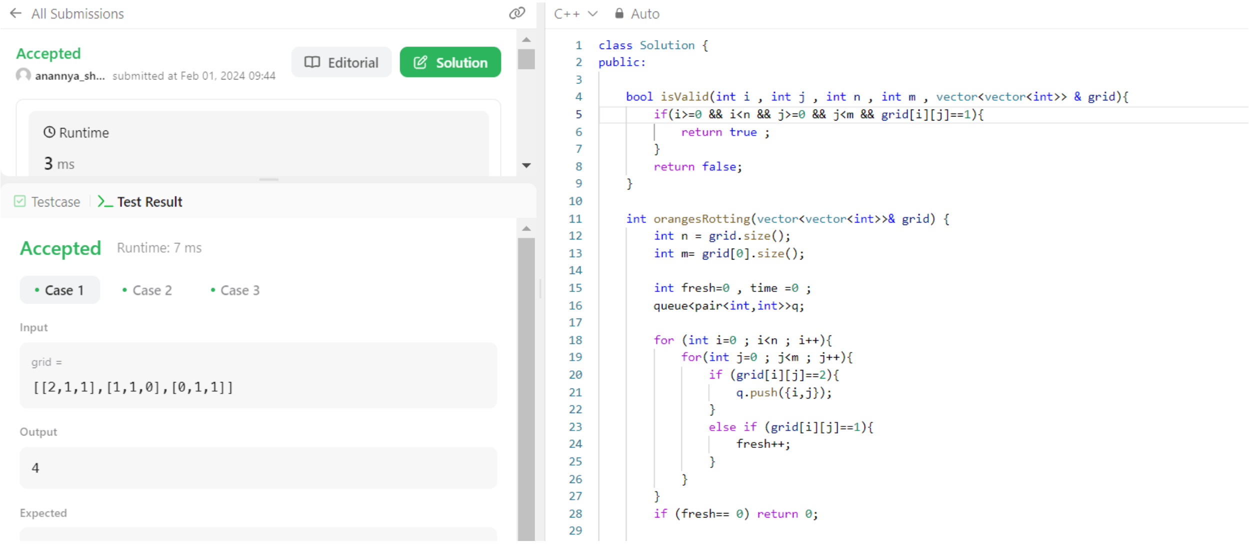
```
1
```

```
class Solution {
    void mark_current_island(vector<vector<char>>& matrix , int x , int y , int r,int c){
        if(x<0||x>=r||y<0 ||y>=c || matrix[x][y]!='1'){
            return;
        }
        matrix[x][y]='2';

        mark_current_island(matrix , x+1 , y , r, c);
        mark_current_island(matrix , x , y+1 , r, c);
        mark_current_island(matrix , x-1 , y , r, c);
        mark_current_island(matrix , x , y-1 , r, c);
    }
public:
    int numIslands(vector<vector<char>>& grid) {
        ios_base::sync_with_stdio(false);
        cin.tie(NULL);

        int rows = grid.size();
        if(rows==0){
            return 0;
        }
        int cols= grid[0].size();

        int no_of_islands = 0 ;
        for(int i=0 ;  i<rows ; ++i){
            for(int j=0 ; j<cols ; ++j){
                if (grid[i][j]=='1'){
                    mark_current_island(grid , i , j , rows , cols);
                    no_of_islands += 1;
                }
            }
```

Saved to local

Ln 3, Col 26

QUES 2



Accepted
anannya_sh... submitted at Feb 01, 2024 09:44

Runtime

3 ms

Accepted Runtime: 7 ms

• Case 1 • Case 2 • Case 3

Input

grid =

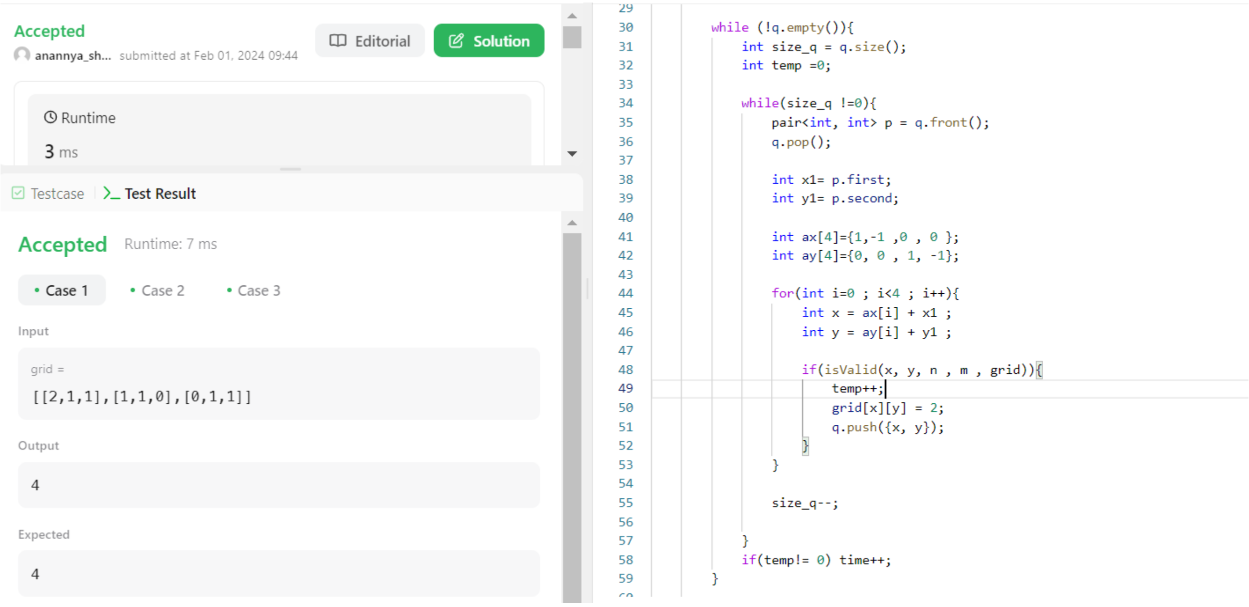
```
[[2,1,1],[1,1,0],[0,1,1]]
```

Output

```
4
```

Expected

```
class Solution {
public:

    bool isValid(int i , int j , int n , int m , vector<vector<int>> & grid){
        if(i>=0 && i<n && j>=0 && j<m && grid[i][j]==1){
            return true ;
        }
        return false;
    }

    int orangesRotting(vector<vector<int>>& grid) {
        int n = grid.size();
        int m= grid[0].size();

        int fresh=0 , time =0 ;
        queue<pair<int,int>>q;

        for (int i=0 ; i<n ; i++){
            for(int j=0 ; j<m ; j++){
                if (grid[i][j]==2){
                    q.push({i,j});
                }
                else if (grid[i][j]==1){
                    fresh++;
                }
            }
        }
        if (fresh== 0) return 0;

```



Accepted
anannya_sh... submitted at Feb 01, 2024 09:44

Runtime
3 ms

Accepted Runtime: 7 ms

• Case 1 • Case 2 • Case 3

Input

grid =
```
[[2,1,1],[1,1,0],[0,1,1]]
```

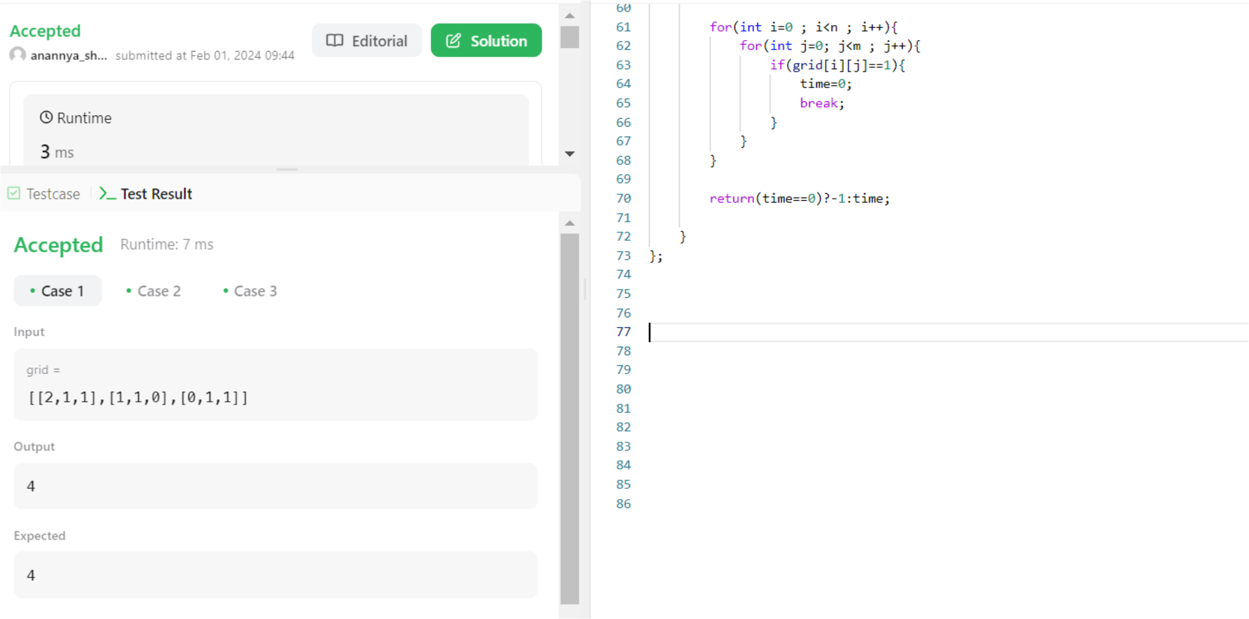
Output

```
4
```

Expected

```
4
```

```
        while (!q.empty()){
            int size_q = q.size();
            int temp =0;

            while(size_q !=0){
                pair<int, int> p = q.front();
                q.pop();

                int x1= p.first;
                int y1= p.second;

                int ax[4]={1,-1 ,0 , 0 };
                int ay[4]={0, 0 , 1, -1};

                for(int i=0 ; i<4 ; i++){
                    int x = ax[i] + x1 ;
                    int y = ay[i] + y1 ;

                    if(isValid(x, y, n , m , grid)){
                        temp++;
                        grid[x][y] = 2;
                        q.push({x, y});
                    }
                }

                size_q--;

            }
            if(temp!= 0) time++;
        }
```



Accepted
anannya_sh... submitted at Feb 01, 2024 09:44

Runtime
3 ms

Accepted Runtime: 7 ms

• Case 1 • Case 2 • Case 3

Input

grid =
```
[[2,1,1],[1,1,0],[0,1,1]]
```

Output

```
4
```

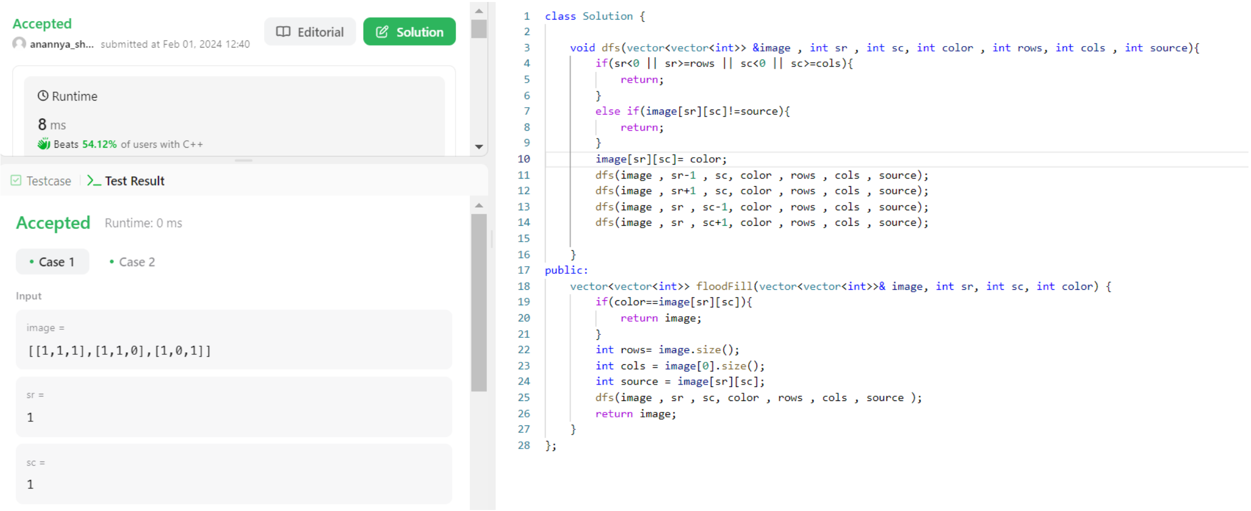
Expected

```
4
```

```

        for(int i=0 ; i<n ; i++){
            for(int j=0; j<m ; j++){
                if(grid[i][j]==1){
                    time=0;
                    break;
                }
            }
        }

        return(time==0)?-1:time;

    }
};
```

QUES 3

QUES 4



Accepted
anannya_sh... submitted at Feb 01, 2024 12:40

Runtime
8 ms
Beats 54.12% of users with C++

Accepted Runtime: 0 ms

Input

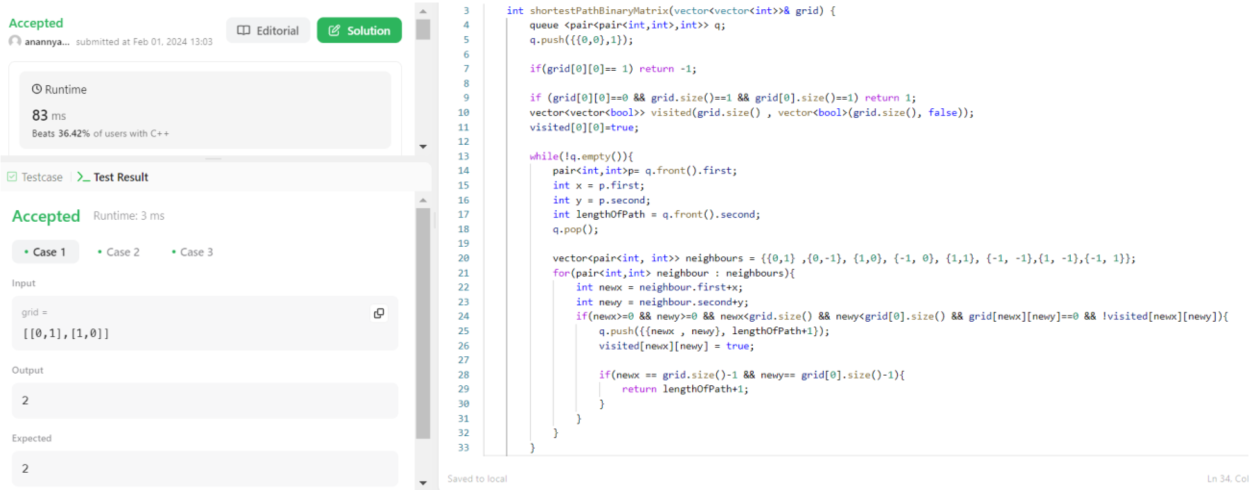
image =
[[1,1,1],[1,1,0],[1,0,1]]

sr =
1

sc =
1

```
class Solution {

    void dfs(vector<vector<int>> &image , int sr , int sc, int color , int rows, int cols , int source){
        if(sr<0 || sr>=rows || sc<0 || sc>=cols){
            return;
        }
        else if(image[sr][sc]!=source){
            return;
        }
        image[sr][sc]= color;
        dfs(image , sr-1 , sc, color , rows , cols , source);
        dfs(image , sr+1 , sc, color , rows , cols , source);
        dfs(image , sr , sc-1, color , rows , cols , source);
        dfs(image , sr , sc+1, color , rows , cols , source);

    }
public:
    vector<vector<int>> floodFill(vector<vector<int>>& image, int sr, int sc, int color) {
        if(color==image[sr][sc]){
            return image;
        }
        int rows= image.size();
        int cols = image[0].size();
        int source = image[sr][sc];
        dfs(image , sr , sc, color , rows , cols , source );
        return image;
    }
};
```

QUES 5



Accepted
anannya... submitted at Feb 01, 2024 13:03

Runtime
83 ms
Beats 36.42% of users with C++

Accepted Runtime: 3 ms

Input

grid =
[[0,1],[1,0]]

Output

2

Expected

2

```
    int shortestPathBinaryMatrix(vector<vector<int>>& grid) {
        queue <pair<pair<int,int>,int>> q;
        q.push({{0,0},1});

        if(grid[0][0]== 1) return -1;

        if (grid[0][0]==0 && grid.size()==1 && grid[0].size()==1) return 1;
        vector<vector<bool>> visited(grid.size() , vector<bool>(grid.size(), false));
        visited[0][0]=true;

        while(!q.empty()){
            pair<int,int>p= q.front().first;
            int x = p.first;
            int y = p.second;
            int lengthOfPath = q.front().second;
            q.pop();

            vector<pair<int, int>> neighbours = {{0,1} ,{0,-1}, {1,0}, {-1, 0}, {1,1}, {-1, -1},{1, -1},{-1, 1}};
            for(pair<int,int> neighbour : neighbours){
                int newx = neighbour.first+x;
                int newy = neighbour.second+y;
                if(newx>=0 && newy>=0 && newx<grid.size() && newy<grid[0].size() && grid[newx][newy]==0 && !visited[newx][newy]){
                    q.push({{newx , newy}, lengthOfPath+1});
                    visited[newx][newy] = true;

                    if(newx == grid.size()-1 && newy== grid[0].size()-1){
                        return lengthOfPath+1;
                    }
                }
            }
        }
```

Saved to local
Ln 34, Col

s