Câu hỏi 4

Chính xác

Chấm điểm của 2,00

Given a n*m grid where each cell in the grid can have a value of 0, 1 or 2, which has the following meaning:

- 1. Empty cell
- 2. This cell contains a fresh apple
- 3. This cell contains a rotten apple

After 1 second, the cell with rotten apple will rot all fresh apples in all the cells adjacent to it (i.e the cells (x+1, y), (x-1, y), (x, y+1), (x, y-1)) Determine the minimum time (in seconds) required to rot all apples. If this cannot be done, return -1.

Note: iostream, vector, and queue are already included.

Constraint:

```
1 <= n, m <= 500
```

Hint: Have you ever heard about breadth-first-search?

```
Example 1:
Input: grid = {{2,2,0,1}}
Output: -1
Explanation:
The grid is
2 2 0 1
```

The apple at (0, 3) cannot be rotten

```
Example 2:
Input: grid = {{0,1,2},{0,1,2},{2,1,1}}
Output: 1
Explanation:
The grid is
0 1 2
0 1 2
2 1 1
```

Apples at positions (0,2), (1,2), (2,0)

will rot apples at (0,1), (1,1), (2,2) and (2,1) after 1 second.

For example:

Test		Result
int rows, cols;	1 4	-1
cin >> rows >> cols;	2 2 0 1	
<pre>vector<vector<int>> grid(rows, vector<int>(cols));</int></vector<int></pre>		
for(int i = 0; i < rows; i++) {		
for(int j = 0; j < cols; j++) cin >> grid[i][j];		
}		
<pre>cout << secondsToBeRotten(grid);</pre>		
int rows, cols;	3 3	1
cin >> rows >> cols;	0 1 2	
<pre>vector<vector<int>> grid(rows, vector<int>(cols));</int></vector<int></pre>	0 1 2	
for(int i = 0; i < rows; i++) {	2 1 1	
for(int j = 0; j < cols; j++) cin >> grid[i][j];		
}		
<pre>cout << secondsToBeRotten(grid);</pre>		

Answer: (penalty regime: 0 %)

```
// iostream, vector and queue are included
    // Hint: use breadth-first-search
 4 v int secondsToBeRotten(vector<vector<int>>& grid) {
 5
        int n = grid.size();
 6
        int m = grid[0].size();
 7
        queue<pair<int, int>> q;
        int freshCount = 0;
 8
 9 🔻
        for (int i = 0; i < n; i++) {
             for (int j = 0; j < m; j++) {
10
                 if (grid[i][j] == 2) {
11
12
                     q.push({i, j});
13
14 🔻
                 else if (grid[i][j] == 1) {
15
                     freshCount++;
16
             }
17
18
19
        int time = 0;
        while (!q.empty()) {
20 🔻
             int size = q.size();
for (int i = 0; i < size; i++) {</pre>
21
22 •
23
                 int x = q.front().first;
24
                 int y = q.front().second;
25
                 q.pop();
                 if (x > 0 \&\& grid[x - 1][y] == 1) {
26 🔻
27
                     grid[x - 1][y] = 2;
28
                     q.push({x - 1, y});
29
                     freshCount--;
30
31
                 if (x < n - 1 \&\& grid[x + 1][y] == 1) {
32
                     grid[x + 1][y] = 2;
                     q.push({x + 1, y});
33
34
                     freshCount--;
35
36 ▼
                 if (y > 0 \&\& grid[x][y - 1] == 1) {
37
                     grid[x][y - 1] = 2;
38
                     q.push({x, y - 1});
39
                     freshCount--;
```

Precheck

Kiểm tra

	Test	Input	Expected	Got	
~	<pre>int rows, cols; cin >> rows >> cols; vector<vector<int>> grid(rows, vector<int>(cols)); for(int i = 0; i < rows; i++) { for(int j = 0; j < cols; j++) cin >> grid[i][j]; } cout << secondsToBeRotten(grid);</int></vector<int></pre>	1 4 2 2 0 1	-1	-1	~
~	<pre>int rows, cols; cin >> rows >> cols; vector<vector<int>> grid(rows, vector<int>(cols)); for(int i = 0; i < rows; i++) { for(int j = 0; j < cols; j++) cin >> grid[i][j]; } cout << secondsToBeRotten(grid);</int></vector<int></pre>	3 3 0 1 2 0 1 2 2 1 1	1	1	~

BÁCH KHOA E-LEARNING



WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

♀ 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

(028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

elearning@hcmut.edu.vn

Copyright 2007-2022 BKEL - Phát triển dựa trên Moodle