

542. 01 Matrix

Add to List

Description Submissions Solutions

- Total Accepted: **3192**
- Total Submissions: **9595**
- Difficulty: **Medium**
- Contributors: [Stomach_ache](#)

Given a matrix consists of 0 and 1, find the distance of the nearest 0 for each cell.

The distance between two adjacent cells is 1.

Example 1:

Input:

```
0 0 0
```

```
0 1 0
```

```
0 0 0
```

Output:

```
0 0 0
```

```
0 1 0
```

```
0 0 0
```

Example 2:

Input:

```
0 0 0
```

```
0 1 0
```

```
1 1 1
```

Output:

```
0 0 0
```

```
0 1 0
```

```
1 2 1
```

Note:

1. The number of elements of the given matrix will not exceed 10,000.
2. There are at least one 0 in the given matrix.
3. The cells are adjacent in only four directions: up, down, left and right.

[Subscribe](#) to see which companies asked this question.

```
class Solution {
public:

    vector<int> dir = {0,1,0,-1,0};

    vector<vector<int>> updateMatrix(vector<vector<int>>& matrix) {
        // 1. initialize
        int m = matrix.size(), n = m ? matrix[0].size() : 0, steps = 1;
        queue<pair<int,int>> que;
        for(int i = 0; i < m; ++i){
            for(int j = 0; j < n; ++j)
                if(matrix[i][j] == 0) que.push({i,j});
        }

        // 2. BFS
        while(!que.empty()){
            int sz = que.size();
            while(sz-->0){
                auto p = que.front();
                que.pop();
                for(int d = 0; d < 4; ++d){
                    int x = p.first + dir[d], y = p.second + dir[d+1];
                    if(x < 0 || x >= m || y < 0 || y >= n || matrix[x][y] <=
0) continue;
```

```

        matrix[x][y] = -steps;
        que.push({x,y});
    }
}
steps++;
}

// 3. reset values
for(int i = 0; i < m; ++i){
    for(int j = 0; j < n; ++j)
        if(matrix[i][j] < 0) matrix[i][j] = -matrix[i][j];
}
return matrix;
}
};

```