Problem 542: 01 Matrix

Problem Information

Difficulty: Medium
Acceptance Rate: 52.75%
Paid Only: No
Tags: Array, Dynamic Programming, Breadth-First Search, Matrix

Problem Description

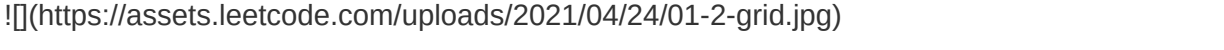
Given an `m x n` binary matrix `mat`, return _the distance of the nearest_`0` _for each cell_.

The distance between two cells sharing a common edge is `1`.

Example 1:

Input: mat = [[0,0,0],[0,1,0],[0,0,0]] **Output:** [[0,0,0],[0,1,0],[0,0,0]]

Example 2:



Input: mat = [[0,0,0],[0,1,0],[1,1,1]] **Output:** [[0,0,0],[0,1,0],[1,2,1]]

Constraints:

* `m == mat.length` * `n == mat[i].length` * `1 <= m, n <= 104` * `1 <= m * n <= 104` * `mat[i][j]` is either `0` or `1`. * There is at least one `0` in `mat`.

Note: This question is the same as 1765: [https://leetcode.com/problems/map-of-highest-peak/](https://leetcode.com/problems/map-of-highest-peak/description/)

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> updateMatrix(vector<vector<int>>& mat) {

}
};
```

Java:

```
class Solution {
public int[][] updateMatrix(int[][] mat) {

}
}
```

Python3:

```
class Solution:
def updateMatrix(self, mat: List[List[int]]) -> List[List[int]]:
```