Problem 3342: Find Minimum Time to Reach Last Room II

Problem Information

Difficulty: Medium
Acceptance Rate: 68.03%
Paid Only: No
Tags: Array, Graph, Heap (Priority Queue), Matrix, Shortest Path

Problem Description

There is a dungeon with `n x m` rooms arranged as a grid.

You are given a 2D array `moveTime` of size `n x m`, where `moveTime[i][j]` represents the **minimum** time in seconds when you can **start moving** to that room. You start from the room `(0, 0)` at time `t = 0` and can move to an **adjacent** room. Moving between **adjacent** rooms takes one second for one move and two seconds for the next, **alternating** between the two.

Return the **minimum** time to reach the room `(n - 1, m - 1)`.

Two rooms are **adjacent** if they share a common wall, either _horizontally_ or _vertically_.

Example 1:

Input: moveTime = [[0,4],[4,4]]

Output: 7

Explanation:

The minimum time required is 7 seconds.

* At time `t == 4`, move from room `(0, 0)` to room `(1, 0)` in one second. * At time `t == 5`, move from room `(1, 0)` to room `(1, 1)` in two seconds.

Example 2:

Input: moveTime = [[0,0,0,0],[0,0,0,0]]

Output: 6

Explanation:

The minimum time required is 6 seconds.

* At time `t == 0`, move from room `(0, 0)` to room `(1, 0)` in one second. * At time `t == 1`, move from room `(1, 0)` to room `(1, 1)` in two seconds. * At time `t == 3`, move from room `(1, 1)` to room `(1, 2)` in one second. * At time `t == 4`, move from room `(1, 2)` to room `(1, 3)` in two seconds.

Example 3:

Input: moveTime = [[0,1],[1,2]]

Output: 4

Constraints:

* `2 <= n == moveTime.length <= 750` * `2 <= m == moveTime[i].length <= 750` * `0 <= moveTime[i][j] <= 109`

Code Snippets

C++:

```
class Solution {
public:
int minTimeToReach(vector<vector<int>>& moveTime) {

}
};
```

Java:

```
class Solution {
public int minTimeToReach(int[][] moveTime) {

}
}
```

Python3:

```
class Solution:
def minTimeToReach(self, moveTime: List[List[int]]) -> int:
```