

Problem 3651: Minimum Cost Path with Teleportations

Problem Information

Difficulty: Hard

Acceptance Rate: 18.90%

Paid Only: No

Tags: Array, Dynamic Programming, Matrix

Problem Description

You are given a $m \times n$ 2D integer array `grid` and an integer `k`. You start at the top-left cell `(0, 0)` and your goal is to reach the bottom-right cell `(m - 1, n - 1)`.

There are two types of moves available:

* **Normal move** : You can move right or down from your current cell `(i, j)` , i.e. you can move to `(i, j + 1)` (right) or `(i + 1, j)` (down). The cost is the value of the destination cell.

* **Teleportation** : You can teleport from any cell `(i, j)` , to any cell `(x, y)` such that `grid[x][y] <= grid[i][j]` ; the cost of this move is 0. You may teleport at most `k` times.

Return the **minimum** total cost to reach cell `(m - 1, n - 1)` from `(0, 0)` .

Example 1:

Input: grid = [[1,3,3],[2,5,4],[4,3,5]], k = 2

Output: 7

Explanation:

Initially we are at (0, 0) and cost is 0.

Current Position | Move | New Position | Total Cost ---|---|---|--- `(0, 0)` | Move Down | `(1, 0)` | `0 + 2 = 2` `(1, 0)` | Move Right | `(1, 1)` | `2 + 5 = 7` `(1, 1)` | Teleport to `(2, 2)` | `(2, 2)` | `7 + 0 = 7` The minimum cost to reach bottom-right cell is 7.

Example 2:

Input: grid = [[1,2],[2,3],[3,4]], k = 1

Output: 9

Explanation:

Initially we are at (0, 0) and cost is 0.

Current Position | Move | New Position | Total Cost ---|---|---|--- `(0, 0)` | Move Down | `(1, 0)` | `0 + 2 = 2` `(1, 0)` | Move Right | `(1, 1)` | `2 + 3 = 5` `(1, 1)` | Move Down | `(2, 1)` | `5 + 4 = 9` The minimum cost to reach bottom-right cell is 9.

Constraints:

* `2 <= m, n <= 80` * `m == grid.length` * `n == grid[i].length` * `0 <= grid[i][j] <= 104` * `0 <= k <= 10`

Code Snippets

C++:

```
class Solution {
public:
    int minCost(vector<vector<int>>& grid, int k) {
        }
};
```

Java:

```
class Solution {
public int minCost(int[][][] grid, int k) {
    }
```

```
}
```

Python3:

```
class Solution:  
    def minCost(self, grid: List[List[int]], k: int) -> int:
```