Problem 3596: Minimum Cost Path with Alternating Directions I

Problem Information

Difficulty: Medium
Acceptance Rate: 70.98%
Paid Only: Yes
Tags: Math, Brainteaser

Problem Description

You are given two integers `m` and `n` representing the number of rows and columns of a grid, respectively.

The cost to enter cell `(i, j)` is defined as `(i + 1) * (j + 1)`.

The path will always begin by entering cell `(0, 0)` on move 1 and paying the entrance cost.

At each step, you move to an **adjacent** cell, following an alternating pattern:

* On **odd-numbered** moves, you must move either **right** or **down**. * On **even-numbered** moves, you must move either**left** or **up**.

Return the **minimum** total cost required to reach `(m - 1, n - 1)`. If it is impossible, return -1.

Example 1:

Input: m = 1, n = 1

Output: 1

Explanation:

* You start at cell `(0, 0)`. * The cost to enter `(0, 0)` is `(0 + 1) * (0 + 1) = 1`. * Since you're at the destination, the total cost is 1.

Example 2:

Input: m = 2, n = 1

Output: 3

Explanation:

* You start at cell `(0, 0)` with cost `(0 + 1) * (0 + 1) = 1`. * Move 1 (odd): You can move down to `(1, 0)` with cost `(1 + 1) * (0 + 1) = 2`. * Thus, the total cost is `1 + 2 = 3`.

Constraints:

* `1 <= m, n <= 106`

Code Snippets

C++:

```
class Solution {
public:
int minCost(int m, int n) {

}
};
```

Java:

```
class Solution {
public int minCost(int m, int n) {

}
}
```

Python3:

```
class Solution:
def minCost(self, m: int, n: int) -> int:
```