DP-S005 (E008)

Solved Challenges 1/2

Back To Challenges List

Collect Max from a Given Cell

ID:11057 Solved By 821 Users

A game has a board with an **RxC** matrix having R rows and C columns containing positive integer values as cell values. A player can start from the cell **S** whose indices are passed as the input. The player can perform the following two navigations after collecting the points in the cell S.
- The player can move to the right cell.
- The player can move to the bottom cell.
The player cannot come back to the previous row or column. The player navigates until he reaches the bottom-right cell. The program must print the maximum points a player can collect from the given RxC matrix as the output.

Boundary Condition(s):
2 <= R, C <= 100
1 <= Each integer value <= 1000

Input Format:
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.
The $(R+2)^{nd}$ line contains two integers representing the indices of the cell S.

Output Format:
The first line contains the maximum points a player can collect from the given RxC matrix.

Example Input/Output 1:
Input:
4 5
4 2 9 6 1
7 9 6 5 4
5 7 3 8 8
7 4 9 9 4
0 1

Output:
44

Explanation:
The navigation of the player to collect the maximum points from the cell **S (0, 1)** is highlighted below.
4 **2** 9 6 1
7 **9** 6 5 4
5 **7** 3 8 8
7 **4 9 9 4**
The maximum points a player can collect is **44** (2+9+7+4+9+9+4).
Hence the output is 44

Example Input/Output 2:
Input:
3 3
70 76 60
18 64 39
45 28 79
1 1

Output:
182

Max Execution Time Limit: 500 millisecs

Ambiance

Python3 (3.x)

Reset

```
rc = list(map(int, input().strip().split()))
r = rc[0]
c = rc[1]

matrix = []
max_matrix = []

for row in range(r):
    matrix.append(list(map(int, input().strip().split())))
    max_matrix.append([0]*c)

src = list(map(int, input().strip().split()))
startrow = src[0]
startcol = src[1]

max_matrix[startrow][startcol] = matrix[startrow][startcol]

for col in range(startcol+1,c):
    max_matrix[startrow][col] = max_matrix[startrow][col-1] +
        matrix[startrow][col]

for row in range(startrow+1,r):
    max_matrix[row][startcol] = max_matrix[row-1][startcol] +
        matrix[row][startcol]

for row in range(startrow+1,r):
    for col in range(startcol+1,c):
        max_matrix[row][col] = max(max_matrix[row-1][col],
            max_matrix[row][col-1]) + matrix[row][col]

print(max_matrix[r-1][c-1])
```

Code did not pass the execution

TestCase ID: 63143

Input:

```
4 5
4 2 9 6 1
7 9 6 5 4
5 7 3 8 8
7 4 9 9 4
0 1
```

Expected Output:

```
44
```

Your Program Output:

```
43
```

Save Run

Run with a custom test case (Input/Output)