1h 20m left

\mathfrak{R}

ALL

(i)

1

2

3

3. Max Path Sum

There is a puzzle using a rectangular grid. The upper left corner is at (row, column) = (0, 0). Each cell contains an integer. The score starts at 0 and is the sum of all the integers in each cell visited as the grid is traversed. Movement begins in either the top or the bottom row and stays within the bounds of the grid. Only 1 cell can be visited per row per direction. Determine the maximum achievable score.

Movement for the two scenarios are as follows:

1. From a cell (i,j) = (0,p), i.e. in the top row:

- (i+1, j-1)(i+1, j)
- o (i+1, j+1)
- 2. From a cell (i,j) = (rows-1,q), i.e. in the bottom row:
 - (i-1, j-1)(i-1, j)
 - (i-1, j+1)

Example

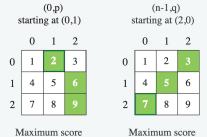
```
board = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

p = 1

q = 0
```

For example, there is a 3x3 grid and the cursor starts at position (0, p) = (0, 1) or (rows - 1, q) = (2, 1). Two possible paths are shown below.

is 7+5+3=15



Each path shown is the highest scoring path from the given position. The better path starts from (0, 2) and the score is 2 + 6 + 9 = 17.

The worse path starts at position (2, 0), where the score is 7 + 5 + 3 = 15.

Function Description

is 2+6+9=17

Complete the function *maxPathSum* in the editor below.

maxPathSum has the following parameter(s):
 int board[n][m]: the values for the grid cells

p: row 0 starting column

q: row n -1 starting column

Returns

int: the maximum achievable score given the two start positions

Constraints

- 2 < n, m < 501
- 0 < board[i][j] < 501.
- $0 \le p, q \le m 1$

► Input Format for Custom Testing

▼ Sample Case 0

Sample Input 0

Sample Output 0

```
Language Python 3
                                                 ( () (?)

    Autocomplete Ready ①
 1 > #!/bin/python3 --
10
11
     # Complete the 'maxPathSum' function below.
12
13
     # The function is expected to return an INTEGER.
14
15
     # The function accepts following parameters:
     # 1. 2D_INTEGER_ARRAY board
16
17
     # 2. INTEGER p
     # 3. INTEGER q
18
19
20
21
     def maxPathSum(board, p, q):
22
         # Write your code here
23
24 > if __name__ == '__main__':--
                                                     Line: 19 Col: 2
  Test
                Custom
                              Run Code
                                           Run Tests
                                                          Submit
 Results
```

Explanation 0The best paths down (left) and up (right) are shown below:

$$(0,p)$$
 - starting at $(0,2)$

	0	1	2
0	9	4	7
1	2	1	3
2	1	4	2

	0	1	2
0		4	7
1	2	1	3
2	1	4	2

Maximum score =
$$7+3+4=14$$

Maximum score = 4+2+9=15