

## 885. Spiral Matrix III

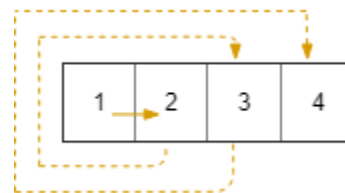
You start at the cell  $(rStart, cStart)$  of an  $rows \times cols$  grid facing east. The northwest corner is at the first row and column in the grid, and the southeast corner is at the last row and column.

You will walk in a clockwise spiral shape to visit every position in this grid. Whenever you move outside the grid's boundary, we continue our walk outside the grid (but may return to the grid boundary later.). Eventually, we reach all  $rows * cols$  spaces of the grid.

Return an array of coordinates representing the positions of the grid in the order you visited them.

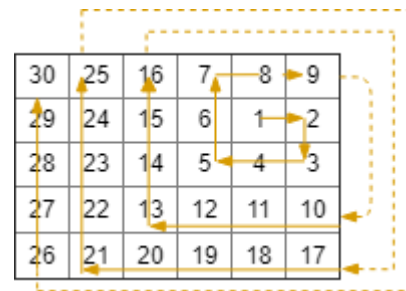
### Example 1:

**Input:**  $rows = 1, cols = 4, rStart = 0, cStart = 0$   
**Output:**  $[[0,0], [0,1], [0,2], [0,3]]$



### Example 2:

**Input:**  $rows = 5, cols = 6, rStart = 1, cStart = 4$   
**Output:**  $[[1,4], [1,5], [2,5], [2,4], [2,3], [1,3], [0,3], [0,4], [0,5], [3,5], [3,4], [3,3], [3,2], [2,2], [1,2], [0,2], [4,5], [4,4], [4,3], [4,2], [4,1], [3,1], [2,1], [1,1], [0,1], [4,0], [3,0], [2,0], [1,0], [0,0]]$



### Constraints:

- $1 \leq rows, cols \leq 100$
- $0 \leq rStart < rows$
- $0 \leq cStart < cols$

## Spiral Matrix III

```
/*
Time complexity: O(n^2)
Space complexity: O(1)
*/
class Solution {
public:
    std::vector<std::vector<int>>> spiralMatrixIII(int rows, int cols, int rStart, int cStart) {
        std::vector<std::vector<int>>> ans;

        std::vector<std::pair<int,int>>> moves {
            {0,1}, //Right
            {1,0}, //Down
            {0,-1}, //Left
            {-1,0} //Up
        };

        int r=rStart,c=cStart,steps=1;

        int dir=0;
        while(ans.size()<rows*cols){
            for(int k=1;k<=2;++k){
                for(int i=1;i<=steps;++i){
                    if(r>=0&&r<rows&&c>=0&&c<cols) ans.push_back({r,c});
                    r+=moves[dir].first;
                    c+=moves[dir].second;
                }
                dir=(dir+1)%4;
            }
            steps+=1;
        }
        return ans;
    }
};
```