

| | | | | |
|----|----|----|----|----|
| 1 | 4 | 7 | 11 | 15 |
| 2 | 5 | 8 | 12 | 19 |
| 3 | 6 | 9 | 16 | 22 |
| 10 | 13 | 14 | 17 | 24 |
| 18 | 21 | 23 | 26 | 30 |

```

minA = 0
maxA = arr[0].len-1
minC = 0
maxC = arr[0].len-1

```

1 4 7 11 15 19 22 24 30 26 23 21 18
10 3 2 5 8 12 16 17 14 13 6 9

```

while (c > 0)
{
    ① For (ci = minc; i <= maxc; i++)
        sys d(arr[cminc] ci)
}

```

$\min r = 1$
 $\max c = 3$
 $\max r = 3$
 $\min c = 1$

```

3 minR++
for (i = minR; i <= maxR; i++) {
    S1S0(a1[r[i]] (maxc))
}
3

```

maxc--

(ii) { for $c_i = \max_{j \geq i} c_j$
 & go on arr(maxn) c_i

3
-XR--

(iv) For $C_i = \max_i i \geq \min_i i \rightarrow ?$
 s.t. $C_i = \max_i i \geq \min_i i$

3 min c++


```

public static void Print(int[][] arr) {
    int minr = 0, minc = 0, maxr = arr.length - 1, maxc = arr[0].length - 1;
    int te = arr.length * arr[0].length; // total_element
    int c = 0;
    while (c < te) {
        for (int i = minc; i <= maxc; i++) {
            System.out.print(arr[minr][i] + " ");
            c++;
            minr++;
        }
        for (int i = minr; i <= maxr; i++) {
            System.out.print(arr[i][maxc] + " ");
            c++;
            maxc--;
        }
        for (int i = maxr; i >= minr; i--) {
            System.out.print(arr[maxr][i] + " ");
            c++;
            maxr--;
        }
        for (int i = maxr; i >= minr; i--) {
            System.out.print(arr[i][minc] + " ");
            c++;
            minc++;
        }
    }
}

```

| | | | | |
|---|----|----|----|---|
| | 0 | 1 | 2 | 3 |
| 1 | 2 | 3 | 4 | |
| 5 | 6 | 7 | 8 | |
| 9 | 10 | 11 | 12 | |

1 2 3 4 8 12 11 10 9 5 6 7 6

Moore Voting Algorithm

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 7 & 3 & -7 & 3 & 7 & 7 & 5 & 7 \end{bmatrix}$$

Given an array `nums` of size `n`, return *the majority element*

The majority element is the element that appears more than $\lfloor n / 2 \rfloor$ times. You may assume that the majority element always exists in the array.

$$m = 7$$

$$n = 7$$

$$[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$$

$$[2 \ 7 \ 3 \ 7 \ 3 \ 7 \ 8 \ 7]$$

2 → 10
 7 → 10
 3 → 10
 7 → 10
 3 → 10
 7 → 10
 2
 3
 3

for (i = 1; i < arr.length; i++) {
 if (arr[i] == ele) {
 vote++
 }
 else {
 vote--
 }
 if (vote == 0) {
 ele = arr[i]
 vote = 1
 }
 }

[illegible]