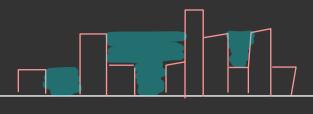
## 0/1/0/2/1/0/1/3/2/1/2/



2mex 0/1/1/2/2/2/2/3/3/3/3

9 max 3/3/3/3/3/3/3/2/2/1

```
public int trap(int[] height) {
   int lmax[] = new int[height.length];
   int rmax[] = new int[height.length];
   lmax[0] = height[0];
   for(int i = 1; i < lmax.length; i++) {
      lmax[i] = Math.max(lmax[i-1], height[i]);
   }
   rmax[rmax.length - 1] = height[rmax.length - 1];
   for(int i = rmax.length - 2; i >= 0; i--) {
      rmax[i] = Math.max(rmax[i + 1], height[i]);
   }
   int ans = 0;
   for(int i = 0; i < height.length; i++) {
      int allowed = Math.min(lmax[i], rmax[i]);
      ans += allowed - height[i];
   }
   return ans;
}</pre>
```

Dur Max of absolute Value Junction n 1/2/3/4 y -1/4/5/6 > (n [i] - n [i] | + (y[i] - y[i] | + (i - j) => n [] - n[] + y [] - y [] + i - j D a[i] +y[i]+i - (a[i] + y[i]+j) 2 - n [i] + y [i] + i - (- n [i] + y [i] + j) 3 xLi) - yCi] + i (4) ールレンーサレンナー

mex 1 - min 1 3

max2 - min2 max max3 - min3 max4 - min4

```
public int maxAbsValExpr(int[] x, int[] y) {
    int max1 = Integer.MIN_VALUE; // -2 ^ 63
    int max2 = Integer.MIN VALUE:
    int max3 = Integer.MIN VALUE;
    int max4 = Integer.MIN VALUE;
    int min1 = Integer.MAX VALUE; // (2 ^ 63) - 1
    int min2 = Integer.MAX VALUE;
    int min3 = Integer.MAX_VALUE;
    int min4 = Integer.MAX VALUE;
    for(int i = 0; i < x.length; i++) {
        \max 1 = \mathsf{Math.max}(\max 1, (x[i] + y[i] + i));
        min1 = Math.min(min1, (x[i] + y[i] + i));
        max2 = Math.max(max2, (-x[i] + y[i] + i));
        min2 = Math.min(min2, (-x[i] + y[i] + i));
        max3 = Math.max(max3, (x[i] - y[i] + i));
        min3 = Math.min(min3, (x[i] - y[i] + i));
        \max 4 = \mathsf{Math.max}(\max 4, (-x[i] - y[i] + i));
        min4 = Math.min(min4, (-x[i] - y[i] + i));
    return Math.max(Math.max(max1 - min1, max2 - min2),
            Math.max(max3 - min3, max4 - min4));
```

hus Spiral matrix 11 maxt 12 13 14 5 mina 11 16 15 6 10 9 8 7

Top wall paint mine mine mexe

D increase minz Right Wall 1) pozint max c mina

N = 4

Start End

1 saduce maxc

1234567

8 9 10 11 12 13 14

19 16

Bottom Wall

1) print maxx maxc mind 2) reduce max or

Left Wall

1 paint mine maxa

1 increase minc

```
public int[][] generateMatrix(int n) {
    int arr[][] = new int[n][n];
    int minr = 0, maxr = n - 1, minc = 0, maxc = n - 1;
    int count = 1;
    while(count \leq n * n) {
        // top wall
        for(int col = minc; col <= maxc; col++)</pre>
            arr[minr][col] = count++;
        minr++;
        // right wall
        for(int row = minr; row <= maxr; row++)</pre>
            arr[row] [maxc] = count++;
        maxc--;
        // bottom wall
        for(int col = maxc; col >= minc; col--)
            arr[maxr][col] = count++;
        maxr--:
        //left wall
        for(int row = maxr; row >= minr; row--)
            arr[row][minc] = count++;
        minc++;
    return arr;
```