

0	1	0	2	1	0	1	3	2	1	2	1
---	---	---	---	---	---	---	---	---	---	---	---



lmax

0	1	1	2	2	2	2	3	3	3	3	3
---	---	---	---	---	---	---	---	---	---	---	---

rmax

3	3	3	3	3	3	3	3	2	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;
}
```

Ques Max of absolute value function

x

1	2	3	4
---	---	---	---

y

-1	4	5	6
----	---	---	---

$$\Rightarrow \underbrace{|x[i] - x[j]|}_2 + \underbrace{|y[i] - y[j]|}_2 + \underbrace{|i - j|}_2$$

$$\Rightarrow x[i] - x[j] + y[i] - y[j] + i - j$$

① $x[i] + y[i] + i - (x[j] + y[j] + j)$

② $-x[i] + y[i] + i - (-x[j] + y[j] + j)$

③ $x[i] - y[i] + i$

④ $-x[i] - y[i] + i$

$$\left. \begin{array}{l} \text{max1} - \text{min1} \\ \text{max2} - \text{min2} \\ \text{max3} - \text{min3} \\ \text{max4} - \text{min4} \end{array} \right\} \text{max}$$

```
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++) {  
        max1 = Math.max(max1, (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));  
  
        max4 = Math.max(max4, (-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));  
}
```

Ques Spiral matrix II

$n = 4$

maxr

minr

	1	2	3	4
	12	13	14	5
	11	16	15	6
	10	9	8	7

minc
maxc

start End

Top wall

- ① print minr minc maxc
- ② increase minr

Right wall

- ① print maxc minr maxr
- ② reduce maxc

Bottom wall

- ① print maxr maxc minc
- ② reduce maxr

Left wall

- ① print minc maxr minr
- ② increase minc

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16

```
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 <= n * n) {  
        // top wall  
        for(int col = minc; col <= maxc; col++)  
            arr[minr][col] = count++;  
        minr++;  
  
        // right wall  
        for(int row = minr; row <= maxr; row++)  
            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;  
}
```