Agenda :

1. Rain water trapping
2. a. Boundary elements
   b. spiral printing .
3. Next permutation .

Q.

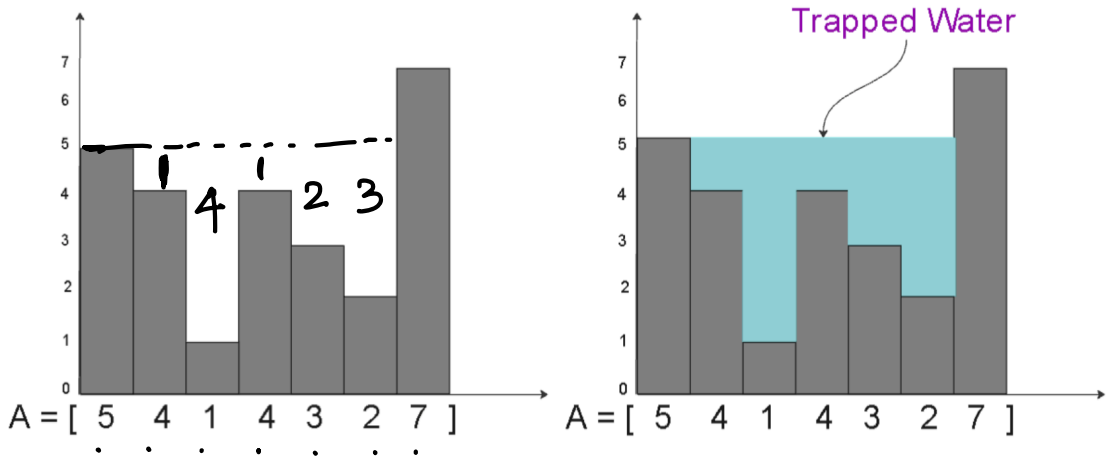
1 sec → <= 10^8 iterations

Imagine a histogram where the bars' heights are given by the array **A**. Each bar is of **uniform width**, which is **1 unit**. When it rains, water will accumulate in the valleys between the bars.

Your task is to calculate the **total** amount of water that can be **trapped** in these valleys.

Example:

The Array A = [5, 4, 1, 4, 3, 2, 7] is visualized as below. The total amount of rain water trapped in A is **11**.



11

Problem Constraints

N = 10^5

1 <= |A| <= 10^5

0 <= A[i] <= 10^5

Target TC = N log N

10^5 → N^2 → 10^{10}

N

$\sqrt{N}$

log N

$N = 10^5$ $\quad <= 10^8$

$N^2 \longrightarrow 10^{10}$ ✗

$N \log N \rightarrow 10^5 * \log_2 10^5 \approx 10^6 - 10^7$ ✓

$N\sqrt{N} \rightarrow 10^5 * \sqrt{10^5} = 10^5 * 10^{5/2}$

$\approx 10^7$

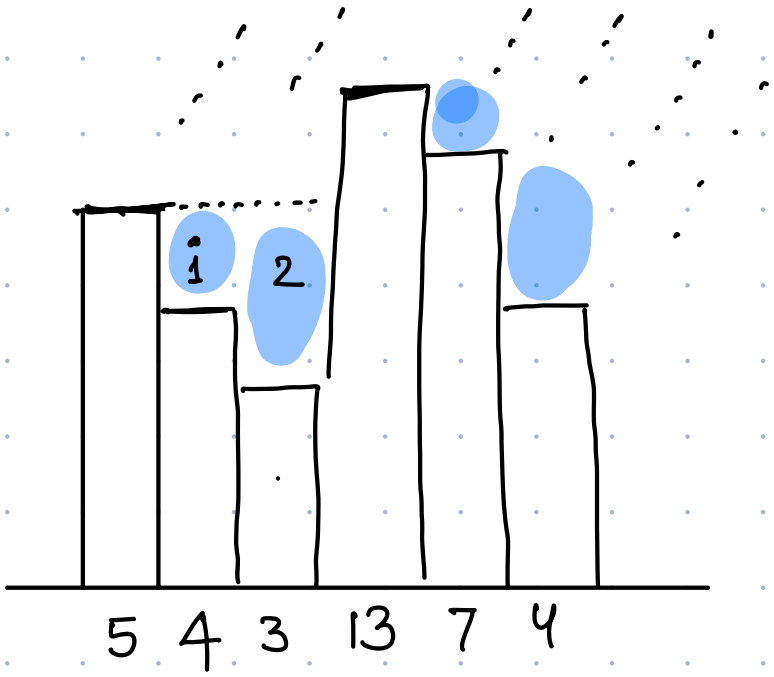
✓ $N \log N$ · B $\quad <= 10^8$ it

✓ N · $\quad <= 10^8$ it

Target TC

$\log N < \sqrt{N} < N < \underbrace{N \log N}_{<=10^8} < N\sqrt{N} < N^2 < N^3 < \underbrace{2^N}_{N=} <= 10^8$

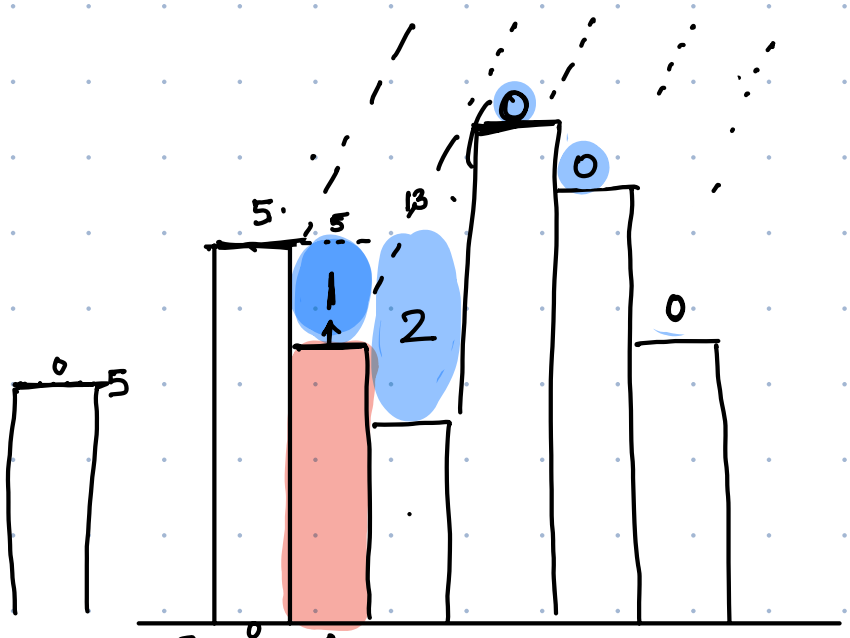
9:17 - 9:25



= 3

→ Vansh, Deepak – same approach

→

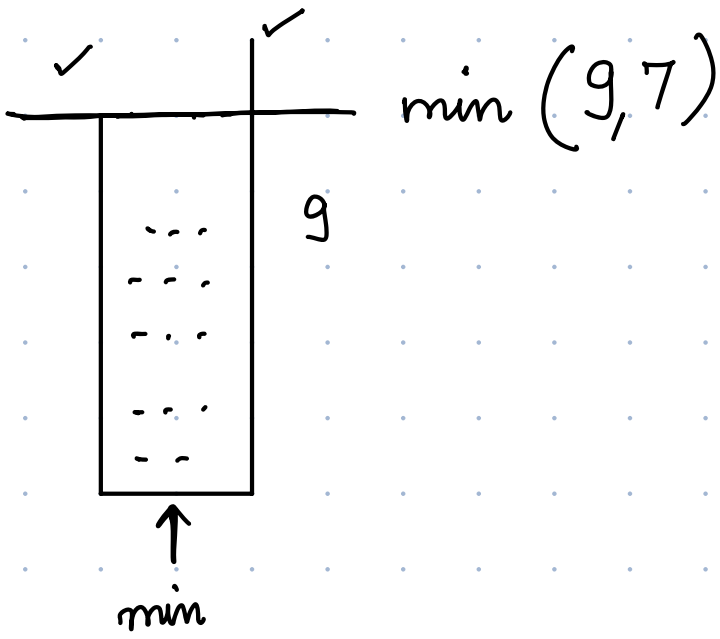


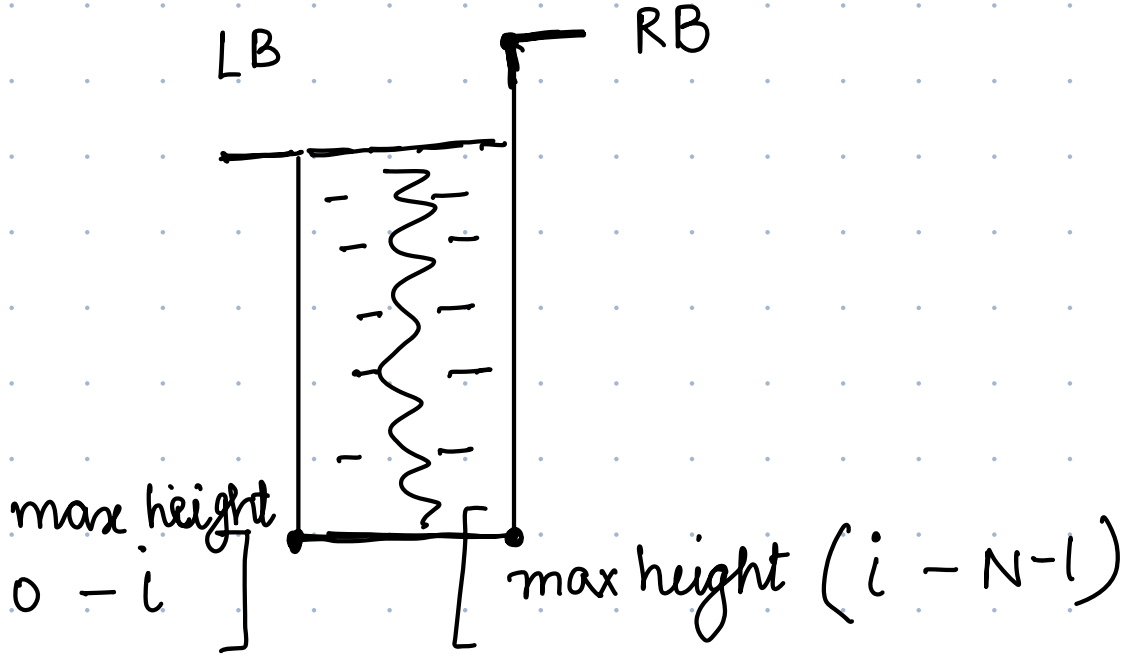
arr[i] 5 4 3 13 7 4

min → 5 5 5 13 7 4

leftMax [5 5 5 13 13 13]
0 1 2 3 4 5

[13 13 13 13 7 4] rightMax
0 1 2 3 4 5



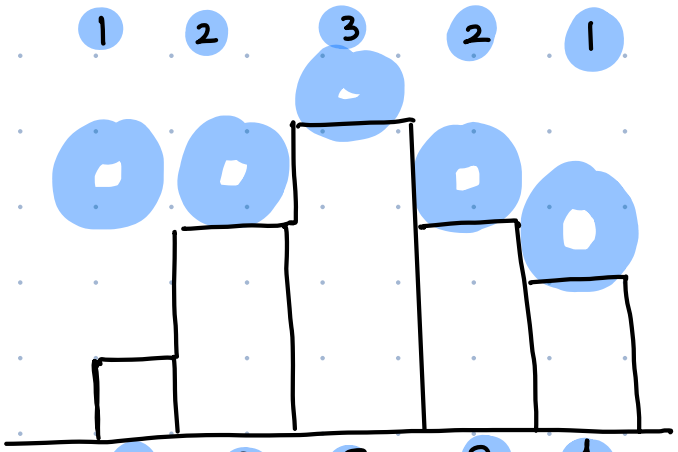


leftMaxArray

RightMax Array

minboundary

wc = arr[i] - minboundary



= 0

minbound

LM 1 2 3 3 3

3 3 3 2 1 RM

```
int n = A.length;
int left[] = new int[n];
int right[] = new int[n];
int maxL = A[0];
int maxR = A[n-1];
int ans = 0;
for(int i = 0 ; i < n ; i++) {
    maxL = Math.max(A[i],maxL);
    left[i] = maxL;
}
for(int i = n-1; i >= 0 ; i--) {
    maxR = Math.max(A[i],maxR);
    right[i] = maxR;
}
for(int i = 1 ; i < n-1 ; i++) {
    int height = Math.min(left[i],right[i]);
    ans += (height - A[i]);
}
```

0 − i = left[i]

i − N-1 = right[i]

N

N

0 0

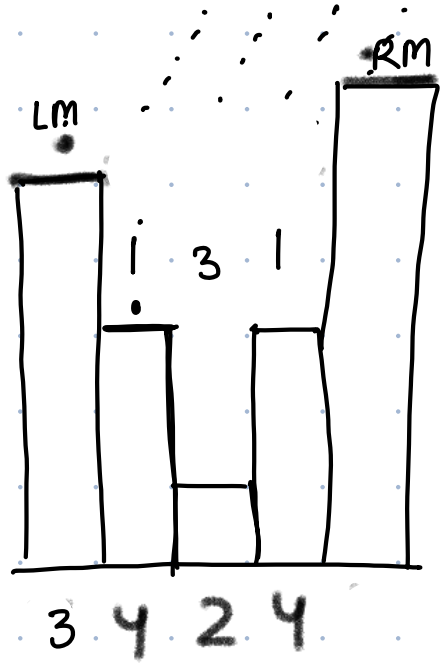
i=0 i=N-1

N

TC = O(N)

SC = O(N)

return ans



$= 5$

$\min(LM, RM)$

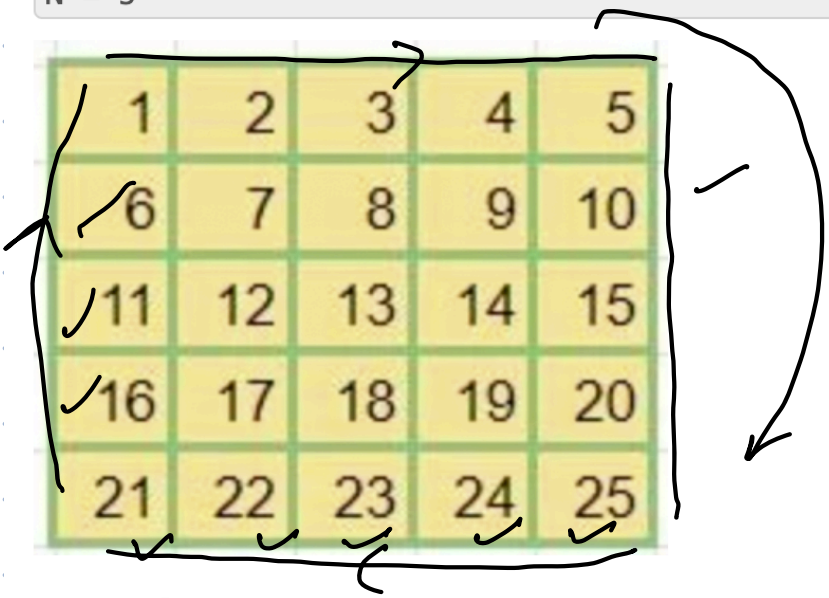
Q2a Boundary elements.

Problem Statement

Given an matrix of N X N i.e. Mat[N][N], print boundary elements in clockwise direction.

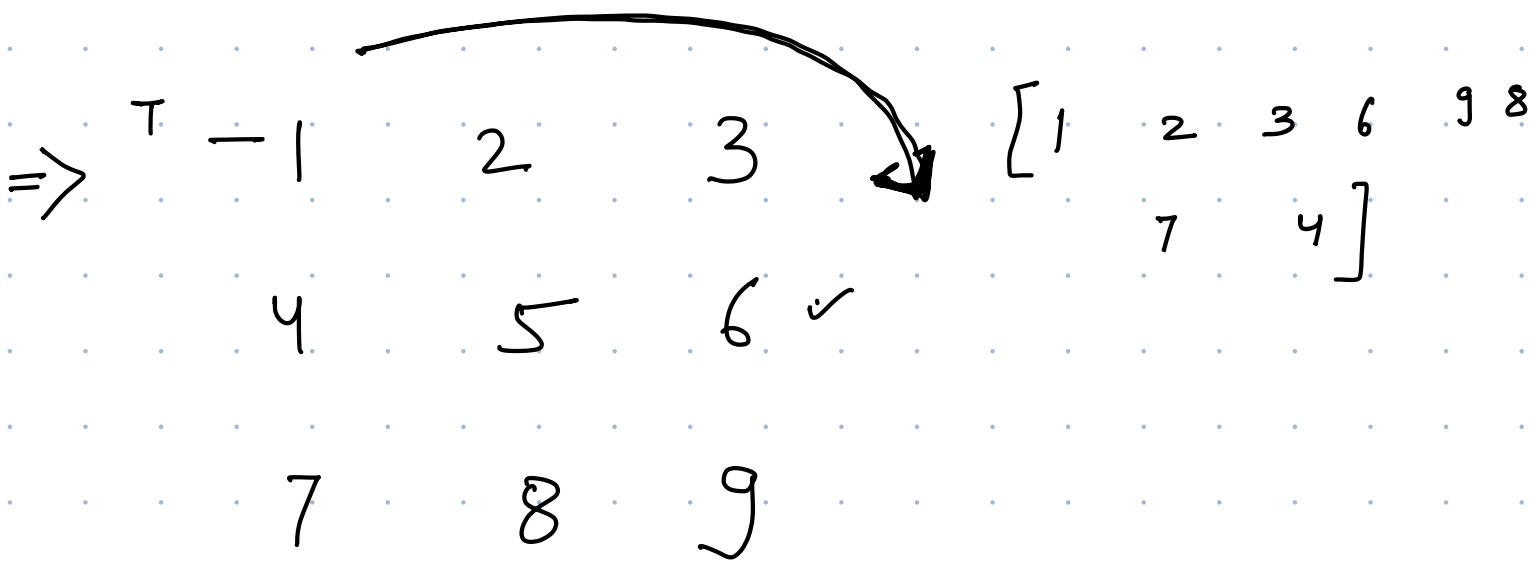
Example:

```
N = 5
```

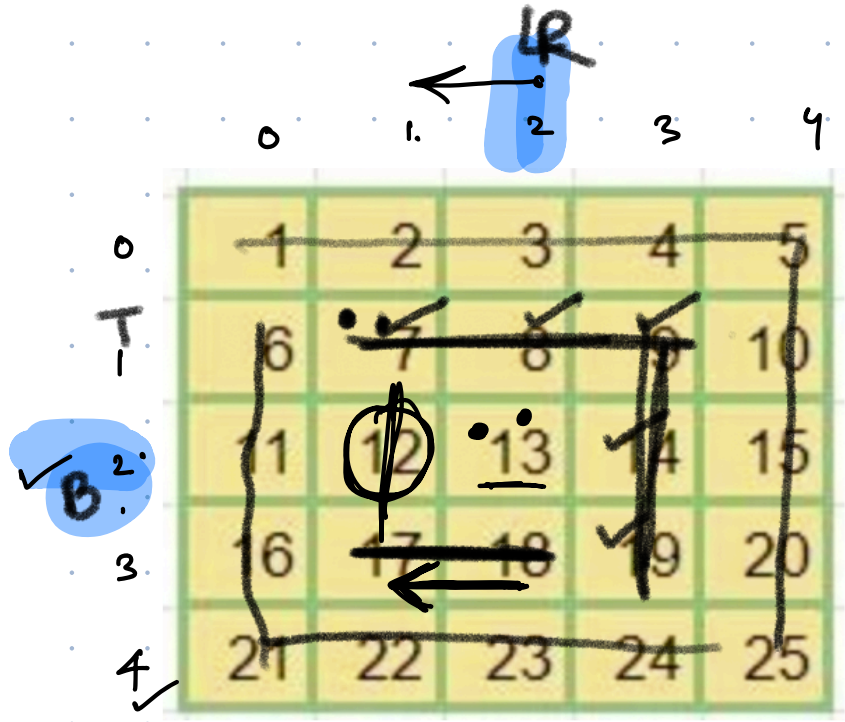


N*N Square matrix.

Output: [1, 2, 3, 4, 5, 10, 15, 20, 25, 24, 23, 22, 21, 16, 11, 6]



T
R
B
L



mat[T][L]

T = 0

R = 4

B = 4

L = 0

Row col

T L to R

T to B R

B R to L col--

B to T L

row--

5x5

T = 0 , B = N-1

L = 0 , R = N-1

```
row = T
for ( col = L ; col <= R; col++){
    print (mat[row][col])
}
T++
col = R
for ( row = T  row <= B  row++)
{
    print (mat[row][col])
}
R--
```

```
row = B
for ( col = R    col >= L  col-- ){
        print ( mat [row][col]

}
B--

col = L
for ( row = B  row >= T  row-- ){
        print ( mat[row][col]

}
L++
```

```
T = 0 , B = N-1        L = 0 , R = N-1

while (L <= R && T <= B ) {

    row = T
    for ( col = L ; col <= R; col++){
        print (mat[row][col])

    }
    T++
    col = R
    for( row = T   row <= B   row++)
    {
        print (mat[row][col])

    }
    R --
    row = B
    for ( col = R    col >= L  col--){
        print (mat [row][col]

    }
    B--

    col = L
    for( row = B  row >= T  row--){
        print (mat[row][col])

    }
    L++

}
```

```
if ( T == B && L == R)
    print ( mat[T][L] );
```

Observations ← Examples

Observations → Code ✓

Examples → 4-5 [10 examples]

Input ⟶ Output

10 mins → break

Array A of size N

digits = arr[i] => 0 to 9

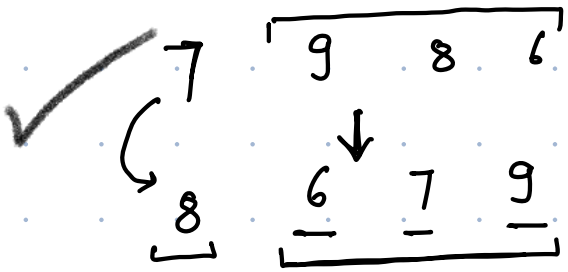
[2, 1, 3] ⟶

[2, 3, 1]

1, 2, 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

Test as custom input: [4, 9, 1, 5] → just greater permutation → 4951

7, 9, 8, 6 → 8679



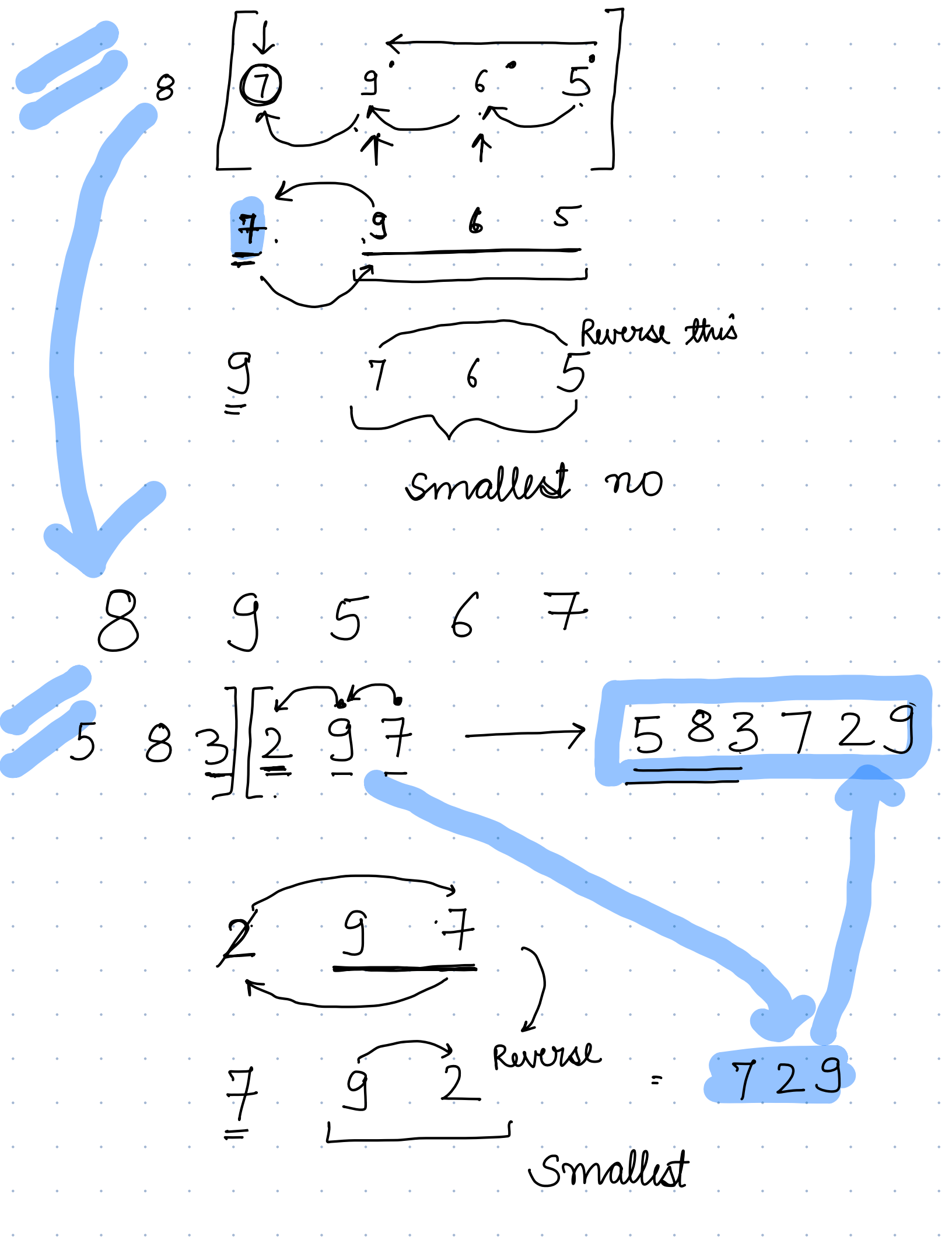
6 9 7

9 5 6 7

8, 7, 9, 6, 5 → 89567

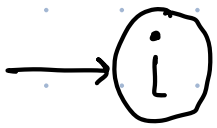
5 8 3 2 9 7 → 5 8 3 7 2 9

4 7 1 2 3 → 4 7 1 3 2


8
7 9 6 5
7 9 6 5
9
7 6 5
Reverse this
Smallest no
8 9 5 6 7
5 8 3
2 9 7
5 8 3 7 2 9
2 9 7
7
9 2
Reverse
= 7 2 9
Smallest

Steps


```
while (i > 0) {
    if (A[i-1] >= A[i])
        i--
    else
        break;
}
```



swap (i-1) with just greater element [i to N-1]

(g)

⇒ swap (i-1 with g) index

⇒ Reverse [i to N-1]

9 7

9 6 5

desc

5 6 7

asc