

Given an array, print the Next Greater Element (NGE) for every element. The Next Greater Element for an element  $x$  is the first greater element on the right side of  $x$  in array. Elements for which no greater element exist, consider next greater element as -1.

next greater  
next smy  
prev greater  
prev smy

$[5, 0, 5, 3, 2, 8, 7, 9, 11, 3]$  (N2)

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

-1 8 8 8 9 3 11 11

$5 > 50$   
 $arr[i] > arr[j]$

↓ ↓ ↓

$[5, 0, 5, 3, 2, 8, 7, 9, 11, 3]$

↑

$for(i=0; i<arr.length; i++)$

$int x = -1$

$for(j=i+1; j<arr.length; j++)$

$if(arr[j] > arr[i])$

$x = arr[j]$

$if(j == arr.length - 1)$

$x = -1$

$ans[i] = x$

$5 > arr[i]$

$arr[i] > arr[j]$

$for(i=0; i<arr.length; i++)$

$while(!st.empty() && arr[i] > arr[st.peek()])$

$ans[st.pop()] = arr[i]$

$st.push(i)$

8 3

7 11

0 50

0 1 2 3 4 5 6 7 8

8 8 8 9 3 11 11

5

30

35

40

38

35

Output

12 3 1 1 END

Daily Temp

Sunny bulbs

30 35 40 38 35

↓ ↓ ↓ ↓ ↓

1 2 3 1 1

20 25 40 38 35

1 2 3 1 1

$[5, 0, 5, 3, 2, 8, 7, 9, 11, 3]$

↑ ↑

0 1 2 3 4 5 6 7 8

1 1 1 1 4 1 6 8

0 1 2 3 4 5 6 7 8

7

$O(n)$

$for(i=0; i<arr.length; i++)$

$while(!st.empty() && arr[i] > arr[st.peek()])$

$st.pop()$

$if(st.empty())$

$ans[i] = -1$

$else$

$ans[i] = i - arr[st.peek()]$

$st.push(i)$

2 3 5 4 6 1 7

next smy

prev smy

$[2, 3, 5, 4, 6, 1, 7]$

$2 \times 5 = 10$

$3 \times 4 = 12$

$5 \times 1 = 5$

$4 \times 6 = 24$

$6 \times 1 = 6$

$1 \times 7 = 7$

$7 \times 1 = 7$

↓ ↓

$[2, 3, 5, 4, 6, 1, 7]$

$R = arr$

$h = 5$

$R = 3, 1, 2, 1$

$8 \times (3 - 1) = 16$

$h = 6$

$R = 5, 1, 2, 1$

$6 \times (5 - 1) = 24$

$h = 2$

$R = 5$

$4 \times 6 = 24$

$h = 4$

$R = 5, 1, 2, 1$

$4 \times (5 - 1) = 16$

$7 \times (7 - 1) = 42$

$(1 \times 7) = 7$

$for(i=0; i<arr.length; i++)$

$area = 5 \times 6 \times 12$

$S < 3$

$while(!st.empty() && arr[i] > arr[st.peek()])$

$int h = arr[st.pop()]$

$int R = i$

$if(st.empty())$

$area = max(area, h \times R)$

$else$

$area = max(area, h \times (R - st.peek()))$

$st.push(i)$

1 0 1 0 0

1 0 1 1 1

1 1 1 1 1

1 0 0 1 0

$int ans = 0;$

$int[] arr = new int[matrix[0].length];$

$for(int i = matrix.length - 1; i >= 0; i--)$

$for(int j = 0; j < matrix[0].length; j++)$

$if(matrix[i][j] == '1')$

$arr[j]++;$

$else$

$arr[j] = 0;$

$ans = Math.max(ans, MaximumArea(arr));$

$1 0 1 0 0$