

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.

8
7
0

3
11
50

↓ ↓ ↓ ↓ ↓ ↓ ↓

[50, 5, 3, 2, 8, 7, 9, 11, 3]

0 1 2 3 4 5 6 7 8

1 8 8 9 9 11 11

0 1 2 3 4 5 6 7 8

For i=0 i<n i+=2

while (!st.isEmpty() && arr[i] > arr[st.peek()]) {

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

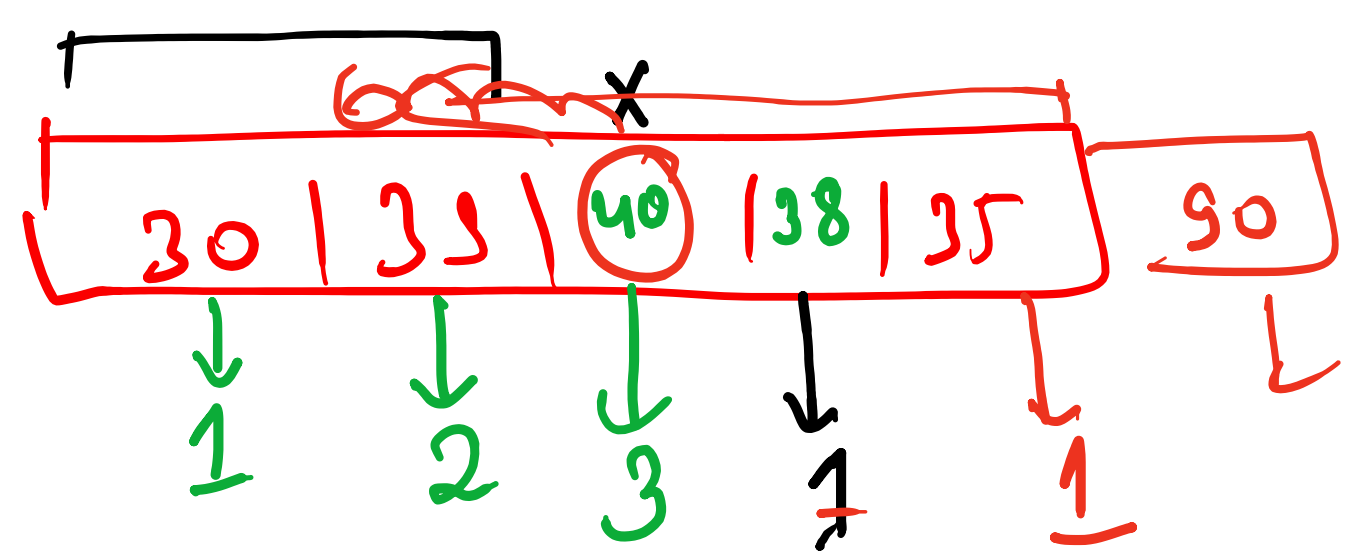
}

st.push(i)

The stock span problem is a financial problem where we have a series of N daily price quotes for a stock and we need to calculate span of stock's price for all N days. You are given an array of length N, where ⁱth element of array denotes the price of a stock on ⁱth. Find the span of stock's price on ⁱth day, for every 1<=i<=N.

A span of a stock's price on a given day, i, is the maximum number of consecutive days before the (i+1)th day, for which stock's price on these days is less than or equal to that on the ⁱth day.

Input
5
30
35
40
38
35
Output
1 2 3 1 1 END



↓ ↓ ↓ ↓ ↓

[50, 5, 3, 2, 8, 7, 9, 11, 3]

0 1 2 3 4 5 6 7 8

1 1 1 1 4 1 6 7 1

0 1 2 3 4 5 6 7 8

For i=0 i<n i+=2

while (!st.isEmpty() && arr[i] > arr[st.peek()]) {

st.pop()

1) Span cal. If (st.isEmpty()) ans[i]=i+1;

else ans[i]=i-st.peek();

st.push(i)

}

2x5=10

3x4=12

5x1=5

4x3=12

6x1=6

1x7=7

7x1=7

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

i=0 → 1x2 = 2

i=1 → 3x(2-1)=3

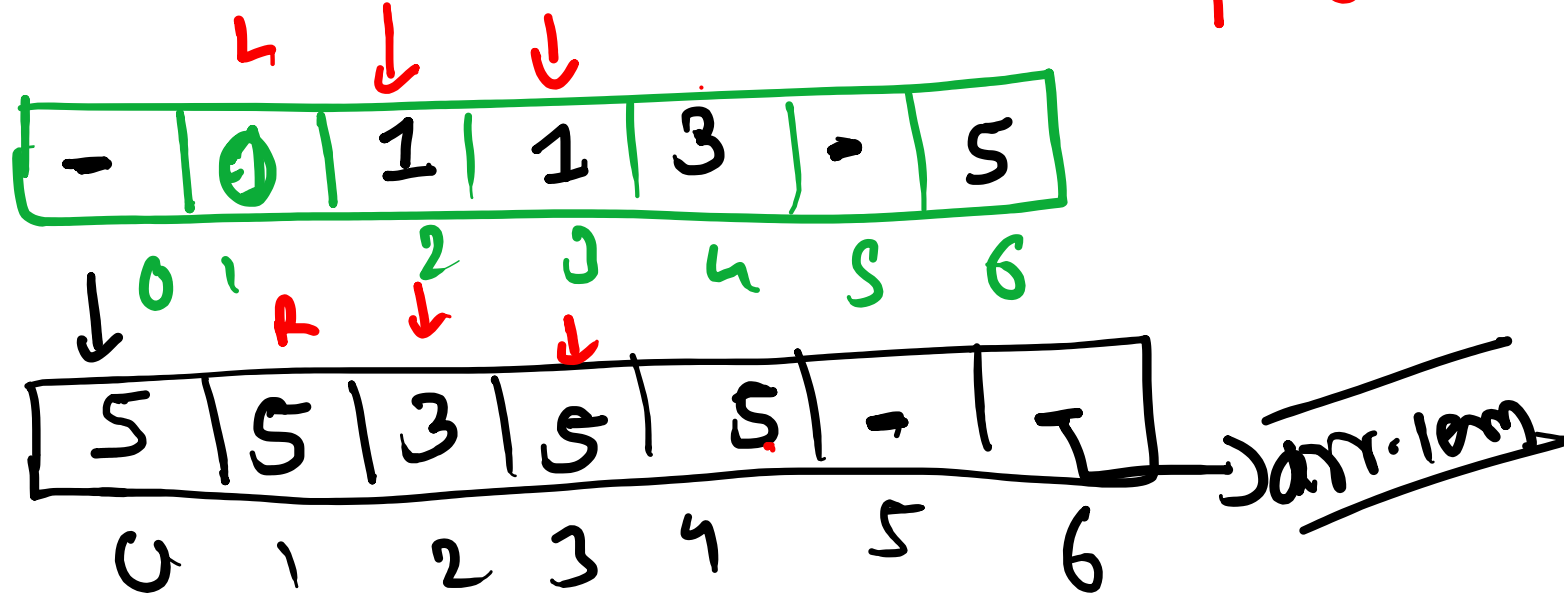
i=2 → 5x(2-1)=5

i=3 → 4x(2-1)=4

i=4 → 6x(2-1)=6

i=5 → 1x(7)=7

i=6 → 7x(7-1)=42



R=7

7x(7-1)=42

1x7=7

6 7

5 4 6 1 7

0 1 2 3 4 5 6

ans=12

5x(3-1)=10

6x(5-3)=12

4x(5-1)=16

3x(5-0)=15

2x(5)=10

For i=0 i<n i+=2

while (!st.isEmpty() && arr[i] < arr[st.peek()]) {

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

int R=i

if (st.isEmpty()) {

area=hxR

ans=max(area,ans)

}

else {

h=st.peek();

area=hx(R-L-1)

ans=max(area,ans)

}

st.push(i)

}

1 1 0 1 0 0

0 1 2 3 4 5

4

6

5 6 3 3 5

1 4 0 0 0 0

0 1 2 3 4

1 1 1 1 1

0 1 2 3 4

→ 1 0 1 0 0

→ 1 0 1 1 1

→ 1 1 1 1 1

1 0 0 1 0

public static void main(String[] args) {

// TODO Auto-generated method stub

char[][] arr = { { '1', '0', '1', '0', '0' }, { '1', '0', '1', '1', '1' }, { '1', '1', '1', '1', '1' }, { '1', '0', '0', '1', '0' } };

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

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

for (int j = 0; j < arr[0].length; j++) {

matrix[i][j] = arr[i][j] - 48;

}

}

int[] a = matrix[matrix.length - 1];

int ans = Maximum_Area(a);

for (int i = matrix.length - 2; i >= 0; i--) {

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

a[j] = matrix[i][j] == 1 ? a[j] + 1 : 0;

}

}

}

4 0 3 0 0

0 1 2 3 4