

Next Greater Element

Given an array of integers, for each element in the array, find the **next greater element** (NGE) in the array. The next greater element for an element x is the first element y to the right of x in the array such that $y > x$. If no such element exists, the NGE for that element is -1 .

012345678

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

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

-1 8 8 8 9 9 11 -1 -1

012345678

5 0 5 3 2 8 7 9 11 3

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

8 8 8 9 9 11 -1 -1

012345678

8 8 8 9 9 11 3

0 1 2 3 4 5 6 7 8

8

7

0

for (i=0; i<n; i++) {

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

arr[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 i^{th} element of array denotes the price of a stock on i^{th} day. Find the span of stock's price on i^{th} day, for every $1 \leq i \leq N$.

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

012345678910

[5, 0, 5, 3, 2, 8, 7, 9, 11, 3, 9, 0, 6]

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

1 1 1 1 4 1 6 7 1 10 1

0123456

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

2

3

5

4

6

1

7

2x5=10

3x4=12

5x1=5

4x3=12

6x1=6

1x7=7

7x1=7

4x(6-1)=12

ux(5-1)=12

for (i=0; i<n; i++) {

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

st.pop();

}

// span cal

i = (st.isEmpty() ? 0 : arr[st.peek()]);

ans[i] = i + 1;

}

are a = 8 & 12

01234567

[2, 3, 5, 4, 6, 1, 7, 0]

↓ ↓ ↓ ↓ ↓ ↓

1 1 1 1 4 1 6 7

0123456

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

2

3

5

4

6

1

7

for (i=0; i<n; i++) {

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

h = arr[st.pop()];

R = i;

if (st.isEmpty()) {

area = max(area, h * R);

}

else {

L = st.peek();

area = max(area, h * (R - L + 1));

}

st.push(i);

}

10100

10111

11111

10010

401300

11111

40

DD

22

742

532

421

133

1234

7→8

1+8

2+8

0123456

DDDDDD

↓ ↓ ↓ ↓ ↓ ↓

32165487

01234567

3214657

0123456

1432768

0123456

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

if (i==s.length() || s.charAt(i) == 'I') {

ans[i] = c; c++;

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

st.pop();

}

st.push(i);

}