

$S \gg S0$ $6 \rightarrow 9$

$[S0, 5, 3, 2, 8, 7, 9, 11, 3]$

$\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$

$-1 \ 8 \ 8 \ 8 \ 9 \ 9 \ 11 \ -1 \ -1$

$\{ \begin{matrix} 8 \\ 7 \\ 0 \end{matrix} \} S0$

for $i=0$ to $arr.length-1$ $i++$

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

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

$'3$

$\rightarrow st.push(i)!$

$\{ -1 | 8 | 8 | 8 | 9 | 9 | 11 | -1 \}$

$0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8$

$10:00$
 $6:00$

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} . 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 $(i+1)^{th}$ day, for which stock's price on these days is less than or equal to that on the i^{th} day.

5
30
35
40
38
35

Output

1 2 3 1 1 END

$30 | 35 | 40 | 38 | 35$

1 2 3 1 1

19+2=7

1 2 3 4 1

$[S0, 5, 3, 2, 8, 7, 9, 10, 3]$

$\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$

1 1 1 1 4 1 6 8 1

```
for (int i = 0; i < arr.length; i++) {
    while (!st.isEmpty() && arr[i] > arr[st.peek()]) {
        st.pop();
    }
    if (!st.isEmpty()) {
        ans[i] = i - st.peek();
    } else {
        ans[i] = i + 1;
    }
    st.push(i);
}
```

public static void Cal_Span(int[] arr) {
Stack<Integer> st = new Stack<>();
int[] ans = new int[arr.length];
for (int i = 0; i < arr.length; i++) {
while (!st.isEmpty() && arr[i] > arr[st.peek()]) {
st.pop();
}
if (!st.isEmpty()) {
ans[i] = i - st.peek();
} else {
ans[i] = i + 1;
}
st.push(i);
}

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

$0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6$

$2 \times 5 = 10$
 $3 \times 4 = 12$
 $5 \times 1 = 5$
 $4 \times 3 = 12$
 $6 \times 1 = 6$
 $1 \times 7 = 7$
 $7 \times 1 = 7$

(12)

Given an array of integers heights representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram.

2 3 5 4 6 1 7

19 10 12

$R-L-1$

PSE
 $IVSE$

NOCL

$2 \ 3 \ 5 \ 4 \ 6 \ 1 \ 7$

$0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$

$R=5, l=0$
 $h=2$

$5 \times (3-1-1) = 5$
 $6 \times (5-3-1) = 6$
 $4 \times (5-1-1) = 12$
 $3 \times (5-0-1) = 12$
 2×6

$ans = 12$

```
for (int i = 0; i < arr.length; i++) {
    while (!st.isEmpty() && arr[i] < arr[st.peek()]) {
        int R = i;
        int h = arr[st.pop()];
        if (!st.isEmpty()) {
            int l = st.peek();
            area = max(area, h * (R - l - 1));
        } else {
            area = max(area, h * l);
        }
        st.push(i);
    }
}
```

$7 \times (7-5-1) = 7$
 $7 \times (7) = 7$

$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

$(2, 1, 1, 2, 1)$
 $(3, 0, 2, 3, 2)$
 $(4, 0, 3, 0, 0)$

$\begin{bmatrix} 4 & 10 & 13 & 14 & 12 \end{bmatrix}$

$0 \ 1 \ 2 \ 3 \ 4$

```
public static int Rectangle(char[][] matrix) {
    int[] arr = new int[matrix[0].length];
    int area=0;
    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;
            }
        }
        area=Math.max(largestArea(arr), area);
    }
    return area;
}
```

area = 6

$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

(5)