

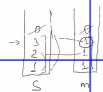
## 2. Реализация стека с операцией минимума

Реализуйте стек с операцией получения минимума. Структура данных должна выглядеть:

- push(x) — добавить элемент в стек
- pop() — удалить верхний элемент
- top() — получить верхний элемент —  $s.top()$
- getMin() — получить минимальный элемент в стеке —  $m.top()$
- empty() — проверить, пуст ли стек

$s, m$

```
def push(x):  
    s.push(x)  
    m.push(min(x, m.top()))  
  
def pop():  
    s.pop()  
    m.pop()  
  
def top():  
    return s.top()
```



$i \cdot u \cdot n \cdot t \cdot m = \min$  от  $q \cdot n \cdot p \cdot o$

## 3. Удаление всех вхождений подстроки

Дана строка  $s$  и подстрока  $t$ . Удалите все вхождения  $t$  в  $s$ , повторяя процесс до тех пор, пока удаления невозможны.

Пример: Вход  $s = aabcbcdc$ ,  $t = abc$   
Выход:  $dc$

Объяснение:  $aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

$aabcbcdc \rightarrow abcdc \rightarrow dc$

## 4. Количество видимых людей в очереди

Даны  $n$  человек в очереди (нумерация от 0 до  $n-1$ ) и массив различ-

ных чисел heights, где heights[i] — рост i-го человека. Человек

i может увидеть человека j ( $i < j$ ), если выполняется

$\min(heights[i], heights[j]) > \max(heights[i+1], heights[i+2], \dots, heights[j-1])$

Верните общее количество, сколько человек увидят хотя бы одного человека в очереди.

Пример: Вход heights = [5, 2, 3, 1, 4]

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

Выход: 4

## 5. Минимальное число операций на подмассиве для формирования целевого массива

Для целочисленного массива target. Также дан целочисленный массив

initial той же длины, где все элементы изначально равны нулю.

За одну операцию можно выбрать любой элемент массива initial

и увеличить каждый его элемент на 1.

Требуется найти минимальное количество операций, необходимых для

получения массива target из массива initial.

Пример 1: Вход target = [1, 2, 3, 2, 1]

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3

Выход: 3