

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 2 (Линейный поиск)

20 мар 2024, 21:12:38
старт: 6 мар 2024, 22:30:00
финиш: 20 мар 2024, 20:00:00
длительность: 13д. 21ч.
начало: 6 мар 2024, 22:30:00
конец: 20 мар 2024, 20:00:00

G. Ни больше ни меньше

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дан массив целых положительных чисел a длины n . Разбейте его на **минимально возможное** количество отрезков, чтобы каждое число было не меньше длины отрезка которому оно принадлежит. Длиной отрезка считается количество чисел в нем. Разбиение массива на отрезки считается корректным, если каждый элемент принадлежит ровно одному отрезку.

Формат ввода

Первая строка содержит одно целое число t ($1 \leq t \leq 1\,000$) — количество наборов тестовых данных. Затем следуют t наборов тестовых данных.
Первая строка набора тестовых данных содержит одно целое число n ($1 \leq n \leq 10^5$) — длину массива.
Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — массив a .
Гарантируется, что сумма n по всем наборам тестовых данных не превосходит $2 \cdot 10^5$.

Формат вывода

Для каждого набора тестовых данных в первой строке выведите число k — количество отрезков в вашем разбиении.
Затем в следующей строке выведите k чисел $len_1, len_2, \dots, len_k$ ($1 \leq len_i \leq n, \sum_{i=1}^k len_i = n$) — длины отрезков в порядке слева направо.

Пример

Ввод <input type="text"/>	Вывод <input type="text"/>
3	3
5	1 2 2
1 3 3 3 2	3
16	1 6 9
1 9 8 7 6 7 8 9 9 9 9 9 9 9 9	3
7	2 3 2
7 2 3 4 3 2 7	

Примечания

Ответы в примере соответствуют разбиениям:

{[1], [3, 3], [3, 2]}

{[1], [9, 8, 7, 6, 7, 8], [9, 9, 9, 9, 9, 9, 9, 9]}

{[7, 2], [3, 4, 3], [2, 7]}

В первом наборе тестовых данных набор длин *{1, 3, 1}*, соответствующий разбиению *{[1], [3, 3, 3], [2]}*, также был бы корректным.

[Скачать условие задачи](#)

Язык Kotlin 1.9.21 (JRE 21)

[Набрать здесь](#)

[Отправить файл](#)

```
1 fun main(args: Array<String>) {
2     val t = readln().toInt()
3
4     val arrays = mutableListOf<List<Int>>()
5
6     for (i in 1..t) {
7         val n = readln().toInt()
8         val array = readln().split(" ").map { it.toInt() }
9         arrays.add(array)
10    }
11
12    for (array in arrays) {
13        val tempRes = mutableListOf<List<Int>>()
14        val tempList = mutableListOf<Int>()
15        var min = array.first()
16
17        array.forEachIndexed { index, num ->
18            if (num >= tempList.size + 1 && min > tempList.size) {
19                tempList.add(num)
20                if (num < min) min = num
21            } else {
22                tempRes.add(tempList.toList())
23                tempList.clear()
24                tempList.add(num)
25                min = num
26            }
27        }
28
29        tempRes.add(tempList)
30        println(tempRes.size)
31        for (list in tempRes) {
32            print(list.size)
33            print(" ")
34        }
35        println()
36    }
37 }
```

[Отправить](#)

[Предыдущая](#)

[Следующая](#)