

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

ОТЧЁТ

по лабораторной работе Разделяй и властвуй (Stepic)

Студент Жумиков Егор группы Р3218

Преподаватель Муромцев Дмитрий Ильич

Санкт-Петербург

2019 г.

Содержание

Задача 1: двоичный поиск	3
Исходный код к задаче 1	3
Задача 2: число инверсий	3
Исходный код к задаче 2	3
Задача 3: точки и отрезки	4
Исходный код к задаче 3	4
Задача 4: сортировка подсчетом	5
Исходный код к задаче 4	5

Задача 1: двоичный поиск

В первой строке даны целое число $1 \leq n \leq 10^5$ и массив $A[1 \dots n]$ из n различных натуральных чисел, не превышающих 10^9 , в порядке возрастания, во второй — целое число $1 \leq k \leq 10^5$ и k натуральных чисел b_1, \dots, b_k , не превышающих 10^9 . Для каждого i от 1 до k необходимо вывести индекс $1 \leq j \leq n$, для которого $A[j] = b_i$, или -1 , если такого j нет.

Sample Input:

```
5 1 5 8 12 13
5 8 1 23 1 11
```

Sample Output:

```
3 1 -1 1 -1
```

Исходный код к задаче 1

```
from bisect import bisect_left

l = list(map(int, input().split()[1:]))
ss = list(map(int, input().split()[1:]))

for s in ss:
    i = bisect_left(l, s)
    print((i + 1) if i < len(l) and l[i] == s else -1, end=' ')
```

Задача 2: число инверсий

Первая строка содержит число $1 \leq n \leq 10^5$, вторая — массив $A[1 \dots n]$, содержащий натуральные числа, не превосходящие 10^9 . Необходимо посчитать число пар индексов $1 \leq i < j \leq n$, для которых $A[i] > A[j]$. (Такая пара элементов называется инверсией массива. Количество инверсий в массиве является в некотором смысле его мерой неупорядоченности: например, в упорядоченном по неубыванию массиве инверсий нет вообще, а в массиве, упорядоченном по убыванию, инверсию образуют каждые два элемента.)

Sample Input:

```
5
2 3 9 2 9
```

Sample Output:

```
2
```

Исходный код к задаче 2

```
from bisect import bisect_left

n = int(input())
a = list(map(int, input().split()))
cnt = 0

def merge(l1, l2):
    global cnt
    l, r = 0, 0
    m = []

    for i in range(len(l1) + len(l2)):
        if l < len(l1) and r < len(l2):
            if l1[l] > l2[r]:
```

```

        cnt += len(l1) - 1
        m.append(l2[r])
        r += 1
    else:
        m.append(l1[l])
        l += 1
    elif r < len(l2):
        m.append(l2[r])
        r += 1
    else:
        m.append(l1[l])
        l += 1

    return m

def msort(a):
    if len(a) == 1:
        return a

    mp = len(a) // 2

    return merge(msort(a[:mp]), msort(a[mp:]))

msort(a)
print(cnt)

```

Задача 3: точки и отрезки

В первой строке задано два целых числа $1 \leq n \leq 50000$ и $1 \leq m \leq 50000$ — количество отрезков и точек на прямой, соответственно. Следующие n строк содержат по два целых числа a_i и b_i ($a_i \leq b_i$) — координаты концов отрезков. Последняя строка содержит m целых чисел — координаты точек. Все координаты не превышают 10^8 по модулю. Точка считается принадлежащей отрезку, если она находится внутри него или на границе. Для каждой точки в порядке появления во вводе выведите, скольким отрезкам она принадлежит.

Sample Input:

```

2 3
0 5
7 10
1 6 11

```

Sample Output:

```

1 0 0

```

Исходный код к задаче 3

```

from bisect import bisect

n, m = map(int, input().split())
bb = []
ee = []
tn = {'b': 0, 'c': 1, 'e': 2}
r = [0] * m

class D:
    def __init__(self, v, t, i=None):
        self.v = v
        self.t = t
        self.p = v * 3 + tn[t]
        self.i = i

class DW:
    def __init__(self, c):
        self._c = c

    def __getitem__(self, i):
        return self._c[i].p

    def __len__(self):

```

```

        return len(self._c)

dwb = DW(bb)
dwe = DW(ee)

for i in range(n):
    a, b = map(int, input().split())
    da = D(a, 'b')
    db = D(b, 'e')

    bb.insert(bisect(dwb, da.p), da)
    ee.insert(bisect(dwe, db.p), db)

for i, d in enumerate(input().split()):
    dd = D(int(d), 'c', i)
    print(bisect(dwb, dd.p) - bisect(dwe, dd.p), end=' ')

```

Задача 4: сортировка подсчетом

Первая строка содержит число $1 \leq n \leq 10^4$, вторая — n натуральных чисел, не превышающих 10. Выведите упорядоченную по неубыванию последовательность этих чисел.

Sample Input:

```

5
2 3 9 2 9

```

Sample Output:

```

2 2 3 9 9

```

Исходный код к задаче 4

```

from collections import Counter
print(*sum([v] * c for (v, c) in sorted(Counter(map(int, [input() for i in
[0,0]][1].split()))).most_common(), key=lambda v:v[0])), [], end=' ')

```