

## 7. Оптимизация алгоритма Мо (mo)

1 сек / 3 сек

60 очков

В процессе подготовки к Межгалактической Олимпиаде по Информатике, Юку обнаружил следующую задачу:

### Суммы в массиве

Дан массив  $A$ , состоящий из  $N$  элементов, а также  $Q$  запросов вида  $(l, r)$ .

В ответ на каждый запрос необходимо посчитать сумму  $A[l] + A[l+1] + \dots + A[r]$ .

Известно, что  $N = Q = 10^5$ .

Юку написал алгоритм решения этой задачи в виде следующего псевдокода:

```
lptr = 1, rptr = 1, sum = A[1], ops = 0
```

```
for each query (ql, qr):
    while rptr < qr:
        rptr += 1
        sum += A[rptr]
        ops += 1
    while lptr > ql:
        lptr -= 1
        sum += A[lptr]
        ops += 1
    while rptr > qr:
        sum -= A[rptr]
        rptr -= 1
        ops += 1
    while lptr < ql:
        sum -= A[lptr]
        lptr += 1
        ops += 1
    print sum
```

К сожалению, время исполнения данной программы превысило ограничения задачи. Юку призадумался, и вспомнил, что на одном из тренировочных лагерей ему что-то рассказывали про какой-то “Алгоритм Мо”, и начал рассуждать: “А не мог бы я упорядочить запросы как-нибудь так, чтобы время исполнения не превышало ограничений. Вообще, насколько быстрой я могу сделать эту программу, если упорядочу запросы неким наилучшим образом?”

Вам даны  $Q$  запросов вида  $(l, r)$ . Вдобавок, гарантируется, что запросы распределены равномерно по массиву (см. замечание ниже). Необходимо упорядочить запросы таким образом, чтобы при исполнении вышеприведённого кода значение переменной `ops` оказалось как можно меньшим.

**Оценивание.** В данной задаче очки выдаются в зависимости от “качества” решения. Программа Юку исполняется на предоставленном упорядочении, и для оценивания используется значение переменной `ops`, полученное в итоге. Решение проверяется на 10 тестах, за

каждый из которых можно получить до 6 очков. Очки за тест считаются по формуле

$$\min\left(6, 6 \cdot \frac{2 \cdot 10^7}{\text{ops}}\right).$$

**Входные данные.** На первой строке текстового файла `moosis.txt` даны два разделённых пробелом целых числа  $N$  и  $Q$  ( $N = Q = 10^5$ ). За ними следует  $Q$  строк, на каждой из которых дан запрос в виде двух разделённых пробелом целых чисел  $l$  и  $r$  ( $1 \leq l \leq r \leq N$ ).

**Выходные данные.** Вывести в текстовый файл `mooval.txt`  $Q$  строк. На каждой строке должен быть выведен запрос в виде двух разделённых пробелом целых чисел  $l$  и  $r$ . Выведенные запросы должны являться перестановкой данных на ввод запросов.

Пример.	<code>moosis.txt</code>	<code>mooval.txt</code>
	7 6	2 7
	1 5	1 2
	2 7	3 6
	4 4	1 2
	1 2	4 4
	3 6	1 5
	1 2	

Обратите внимание, что приведённый в пример вводный файл **не соответствует** условиям задачи, т.к. не выполняется условие  $N = Q = 10^5$ . Пример лишь демонстрирует оформление входных и выходных данных.

**Примечание.** Реальные тестовые данные сгенерированы с помощью следующего кода:

```
print(100000, 100000)
for i in 1...100000:
    l = random(1, 100000)
    r = random(1, 100000)
    if (l < r):
        print(l, r)
    else:
        print(r, l)
```

Функция `random(1, 100000)` в данном коде возвращает равновероятно случайное целое число  $x$  между 1 и 100 000 включительно.