

Тренировочный контекст: разработка бэкенда

27 апр 2023, 20:24:43
старт: 23 апр 2023, 22:03:59
начало: 1 янв 2021, 02:00:00

С. Расстояние

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	2 секунды	512Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	4 секунды	512Mb		
Python 3.7 (PyPy 7.3.3)	4 секунды	512Mb		
Python 2.7	4 секунды	512Mb		
PHP 7.3.5	4 секунды	512Mb		

Рассмотрим целочисленный массив a длины n . Назовём *расстоянием* от индекса i до множества индексов S величину $dist(i, S) = \sum_{j \in S} |a_i - a_j|$.
Зафиксируем целое число k . Рассмотрим функцию $f(i) = \min dist(i, S)$, где минимум берётся по множествам S размера k , **не содержащим индекс i** .
Определите значение $f(i)$ для всех i от 1 до n .

Формат ввода

В первой строке заданы два целых числа n и k ($2 \leq n \leq 300\,000$, $1 \leq k < n$), описанные в условии.
Во второй строке содержится n целых чисел a_i ($1 \leq a_i \leq 10^9$) — элементы массива a .

Формат вывода

Выведите n целых чисел: значения $f(i)$ для $i = 1, i = 2, \dots, i = n$.

Пример 1

Ввод

Вывод

4 2

3 2 2 3

1 2 3 4

Пример 2

Ввод

Вывод

5 3

4 2 8 4 2

3 2 5 1 2

Пример 3

Ввод Вывод 6 2
3 2 1 101 102 103

3 2 3 3 2 3

Примечания

Рассмотрим первый пример.

При $i = 1$ оптимальное S — это $\{2, 3\}$; $\text{dist}(1, \{2, 3\}) = |1 - 2| + |1 - 3| = 3$.

При $i = 2$ оптимальное S — это $\{1, 3\}$; $\text{dist}(2, \{1, 3\}) = |2 - 1| + |2 - 3| = 2$.

При $i = 3$ оптимальное S — это $\{2, 4\}$; $\text{dist}(3, \{2, 4\}) = |3 - 2| + |3 - 4| = 2$.

При $i = 4$ оптимальное S — это $\{2, 3\}$; $\text{dist}(4, \{2, 3\}) = |4 - 2| + |4 - 3| = 3$.

Язык Python 3.7.3

Набрать здесь

Отправить файл

```
1 #import numpy
2 #import time
3 #from numpy import random
4
5 k=0
6 initial_array = list()
7 with open('input.txt') as f:
8     p = f.readline().strip('\n').split(' ')
9     n = int(p[0])
10    k = int(p[1])
11    arr = f.readline().strip('\n').split(' ')
12    i = 0
13    for i in range(n):
14        initial_array.append(int(arr[i]))
15
16 def find_distance(arr):
17     l = list()
18     i = 0
19     for i in range(len(arr)):
20         l.append([arr[i], i])
21     l.sort()
22     dic = dict()
23     moduls = list()
24     i = 0
25     for i in range(len(l)):
26         l_start = i - k
27         sum = 0
28         if l_start < 0:
29             l_start = 0
30         l_pointer = l_start
31         while l_pointer < i:
32             moduls.append(abs(l[i][0] - l[l_pointer][0]))
33             l_pointer += 1
34         r_start = i + k
35         if r_start > len(l) - 1:
36             r_start = len(l) - 1
37         r_pointer = r_start
38         while r_pointer > i:
```

Отправить

Предыдущая

Следующая