

ВШЭ АиСД 2021. Массивы, матрицы

4 дек 2021, 19:01:30

старт: 16 сен 2021, 11:30:00

финиш: 26 сен 2021, 23:59:59

длительность: 10д. 12ч.

начало: 16 сен 2021, 11:30:00

конец: 26 сен 2021, 23:59:59

# А. Вращение массива с максимальным отличием (0.2)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

Вращением массива назовем операцию циклического сдвига элементов одномерного массива вправо (по часовой стрелке).

Расстоянием Хэмминга  $d_H$  между двумя массивами назовем количество позиций, в которых элементы массивов отличаются друг от друга. Например, расстояние Хэмминга  $d_H(A, B)$  между двумя массивами  $A = [1, 3, 1, 4, 5]$  и  $B = [1, 4, 5, 1, 3]$  составляет 4, поскольку значения их элементов различаются во всех позициях, кроме первой.

Нетрудно заметить, что количество возможных вращений массива длины  $N$  составляет  $N$ .

Требуется найти все такие уникальные вращения для заданного одномерного массива, что расстояние Хэмминга между исходным массивом и результатом его вращения максимально.

## Формат ввода

В первой строке – число  $N$ , которое определяет размер массива.  
Гарантируется, что  $N \leq 2 \cdot 10^3$ .

Во второй строке через пробел заданы  $N$  элементов массива.  
Гарантируется, что каждый из них по модулю не превышает  $10^9$ .

## Формат вывода

В первой строке максимальное расстояние Хэмминга, которое можно получить в результате вращения исходного массива.

В последующих строках выводятся **уникальные** вращения массива, которые удалены от исходного на найденное максимальное расстояние Хэмминга **в порядке возрастания** количества элементов, на которые выполняется сдвиг. Через точку запятой от результата вращения массива выводится количество элементов, на которое был выполнен сдвиг его элементов.

## Пример

Ввод	Вывод
4	4
1 1 8 0	8 0 1 1; 2

```
1  /*А. Вращение массива с максимальным отличием*/
2
3  #include <iostream>
4  #include <vector>
5  #include <string>
6
7  std::vector<std::string> PrintShifts(const std::vector<int>& arr, const int& shift,
8                                     std::vector<std::string> allreadyOutput) {
9      std::string output;
10
11     for (unsigned int i = 0; i < arr.size(); ++i) {
12         if (i + 1 != arr.size()) {
13             output += std::to_string(arr[(i + (arr.size() - shift)) % arr.size()]) + " ";
14         }
15     }
16
17     for (unsigned int i = 0; i < allreadyOutput.size(); ++i) {
18         if (output == allreadyOutput[i]) {
19             return allreadyOutput;
20         }
21     }
22
23     allreadyOutput.push_back(output);
24
25     std::cout << output;
26     std::cout << arr[((arr.size() - 1) + (arr.size() - shift)) % arr.size()];
27     std::cout << " " << shift << std::endl;
28
29     return allreadyOutput;
30 }
31
32 std::vector<int> FindDistanceBetweenShifts(const std::vector<int>& arr, std::vector<int> arrShifts,
33                                            const int& shift, int* max) {
34     int counter = 0;
35
36     for (unsigned int i = 0; i < arr.size(); ++i) {
37         if (arr[i] != arr[(i + (arr.size() - shift)) % arr.size()]) {
38
```