

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: М. М. Парфенов  
Преподаватель: С. А. Михайлова  
Группа: М8О-301Б-22  
Дата:  
Оценка:  
Подпись:

Москва, 2025

## Лабораторная работа №8

**Задача:** Разработать жадный алгоритм решения задачи, определяемой своим вариантом.

# 1 Описание

Дана последовательность длины  $N$ , состоящая из целых чисел 1, 2, 3. Необходимо найти минимальное количество обменов элементов последовательности, после которых она станет отсортированной.

## Алгоритм

1. Входные данные содержат: - Количество элементов последовательности  $n$ . - Массив  $arr$  длины  $n$ , содержащий элементы последовательности.
2. Определяем количество каждого числа 1, 2, 3 в последовательности, используя массив  $count$ , где: -  $count[1]$  — количество единиц, -  $count[2]$  — количество двоек, -  $count[3]$  — количество троек.
3. Для минимизации обменов мы обрабатываем элементы в последовательности следующим образом: - Проходим по последовательности и пытаемся расположить все единицы на позиции, предназначенные для них, двоек на позиции для двоек, и троек — для троек. - Если элемент не на своей позиции, то ищем его пару для обмена и производим минимальное количество обменов, чтобы привести последовательность к отсортированному состоянию.
4. После того как все элементы будут на своих местах, количество совершенных обменов будет минимальным.
5. Выводим количество произведённых обменов.

## Сложность

- Временная сложность алгоритма:  $O(n)$ , так как мы проходим по всем элементам массива один раз. - Пространственная сложность:  $O(n)$ , так как нам нужно хранить массив для элементов и массив для подсчёта количества каждого числа.

## 2 Исходный код

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main() {
6     ios::sync_with_stdio(false);
7     cin.tie(0), cout.tie(0);
8     int n;
9     cin >> n;
10    vector<int> arr(n);
11    vector<int> count(4);
12    count[0] = 0;
13    int swap = 0;
14    for(int i = 0; i < n; ++i){
15        cin >> arr[i];
16        count[arr[i]]++;
17    }
18
19    count[3] = count[2] + count[1];
20    count[2] = count[1] + count[0];
21    count[1] = count[0];
22
23    for(int i = 0; i < n; ++i){
24        if(i < count[2]){
25            if(arr[i] == 3){
26                for(int j = n - 1; j >= count[2]; --j){
27                    if(arr[j] == 1){
28                        int tmp_a = arr[i];
29                        arr[i] = arr[j];
30                        arr[j] = tmp_a;
31                        swap++;
32                        break;
33                    }
34                }
35            }
36            else if (arr[i] == 2){
37                for(int j = count[2]; j < n; ++j){
38                    if(arr[j] == 1){
39                        int tmp_a = arr[i];
40                        arr[i] = arr[j];
41                        arr[j] = tmp_a;
42                        swap++;
43                        break;
44                    }
45                }
46            }
47        }
```

```

48      // 2
49      else if(i < count[3]){
50          if(arr[i] != 2){
51              swap++;
52              for(int j = count[3]; j < n; ++j){
53                  if(arr[j] == 2){
54                      int tmp_a = arr[i];
55                      arr[i] = arr[j];
56                      arr[j] = tmp_a;
57                      break;
58                  }
59              }
60          }
61      }
62
63      // for(auto _ : arr){
64      // cout << _ << ' ';
65      // }
66      // cout << '\n';
67  }
68
69  cout << swap;
70  return 0;
71 }

```

### 3 Консоль

```

[±main]-> cat input.txt
3 2 1

```

3

```

[±main]-> g++ main.cpp
[±main]-> ./a.out < input.txt
[±main]->
1

```

## 4 Выводы

Выполнив задачу по нахождению минимального количества обменов для сортировки последовательности из чисел 1, 2, 3, я научился применять жадные подходы для поиска решений. Однако, несмотря на эффективность этого метода для данной задачи, важно понимать, что жадные алгоритмы не всегда приводят к оптимальному решению в более сложных случаях. Перед их применением стоит тщательно анализировать задачу и проверять, гарантирует ли выбранный жадный метод нахождение оптимального решения.

## Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))
- [2] Список использованных источников оформлять нужно по ГОСТ Р 7.05-2008