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: П.А. Земсков

Преподаватель: Н.К. Макаров

Группа: М8О-301Б

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2025

Содержание

1 Лабораторная работа №8

Задача

Требуется разработать программу для поиска образцов в заранее известном тексте с использованием суффиксного массива.
Вариант алгоритма: 5 Оптимальная сортировка чисел

2 Описание

Дана последовательность длины N из целых чисел 1, 2, 3. Необходимо найти минимальное количество обменов элементов последовательности, в результате которых последовательность стала бы отсортированной.

3 Исходный код

Общая идея: Массив состоит только из чисел 1, 2 и 3, поэтому его можно разбить на три зоны, где должны стоять соответствующие числа: сначала все 1, затем 2, потом 3. Считаем, сколько элементов каждого типа находится не на своём месте, и выполняем жадно минимальные обмены: сначала меняем элементы, стоящие в «чужих» зонах (например, 1-2, 1-3, 2-3), а затем устраняем оставшиеся циклические ошибки, каждая из которых требует двух обменов.

Листинг 1: Исходный код

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int N;
    cin >> N;
    vector<int> a(N);
    for (int i = 0; i < N; i++) cin >> a[i];

    int cnt[4] = {0};
    for (int x : a) cnt[x]++;

    int zone1_end = cnt[1];
    int zone2_end = cnt[1] + cnt[2];

    int mis[4][4] = {};

    for (int i = 0; i < N; i++) {
        int zone;
        if (i < zone1_end) zone = 1;
        else if (i < zone2_end) zone = 2;
        else zone = 3;
        mis[zone][a[i]]++;
    }

    int swaps = 0;

    int direct12 = min(mis[1][2], mis[2][1]);
    swaps += direct12;
    mis[1][2] -= direct12;
    mis[2][1] -= direct12;

    int direct13 = min(mis[1][3], mis[3][1]);
    swaps += direct13;
    mis[1][3] -= direct13;
    mis[3][1] -= direct13;

    int direct23 = min(mis[2][3], mis[3][2]);
    swaps += direct23;
    mis[2][3] -= direct23;
    mis[3][2] -= direct23;

    int remaining = mis[1][2] + mis[1][3] + mis[2][1] + mis[2][3] +
        mis[3][1] + mis[3][2];
    swaps += (remaining / 3) * 2;

    cout << swaps << "\n";
```

```
50      return 0;
51  }
```

4 Консоль

Пример компиляции и демонстрация работы программы:

```
C:\Users\jocke\Documents\diskran\lab8> g++ --std=c++20 main.cpp -o main
C:\Users\jocke\Documents\diskran\lab8> ./main
3
3 2 1
1
```

5 Тест производительности

Такой подход даёт минимальное количество перестановок и работает за O(N) времени при O(1) памяти.

Результат:

Количество перестановок чисел, нужных для сортировки.

6 Выводы

В результате выполнения лабораторной работы был реализован жадный алгоритм, определяющий минимальное количество обменов для сортировки последовательности, состоящей из чисел 1, 2 и 3. Использование трёх зон, соответствующих каждому типу элементов, позволило эффективно выявлять и устранять ошибки расположения с помощью прямых и циклических обменов. Алгоритм показал высокую эффективность - линейную сложность по времени и константную по памяти, что подтверждает рациональность выбранного подхода к решению задачи оптимальной сортировки.

7 Список литературы

1. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. *Алгоритмы: построение и анализ. 3-е изд.* – М.: Издательский дом «Вильямс», 2010.

2. Knuth D. E. *The Art of Computer Programming. Vol. 3: Sorting and Searching. 2nd ed.* – Addison-Wesley, 1998.

3. *Жадные алгоритмы — Алгоритмика.* `https://ru.algorithmica.org/cs/combinatorial-op` `greedy/`.