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: И. Д. Черненко
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б

Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №8

Задача: Разработать жадный алгоритм решения задачи, определяемой своим вариантом. Доказать его корректность, оценить скорость и объём затрачиваемой оперативной памяти.

Реализовать программу на языке C или C++, соответсвующую построенному алгоритму. Формат входных и выходных данных описан в варианте задания.

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

Формат входных данных

Число **N** на первой строке и **N** чисел на второй строке.

Формат результата

Минимальное количество обменов.

1 Описание

Жадный алгоритм [1] — алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

В данной задаче введем массив frequency, в котором будем хранить кол-во единиц, двоек и троек(последнее не так важно). Будем проходить по нашей последовательности слева направо, сначала расстваим единицы на их места, потом расставим двойки:

- если видим двойку на позиции с единицей то ищем первую единицу после позиции frequency[0], с тройками аналогично, но с другого конца, чтобы тройки не занимали места двоек
- когда все единицы на своих местах начинаем те же операции но уже между двойками и тройками, при этом начинаем поиск с позиции frequency[0]

После проделанных операций очевидно что на оставшихся позициях останутся тройки.

2 Исходный код

```cpp
#include <iostream>
#include <vector>
int main() {
    int size;
    std::cin >> size;
    std::vector<int> data(size);
    std::vector<int> frequency(3);
    for (int i = 0; i < size; i++) {
        std::cin >> data[i];
        if (data[i] == 1) {
            frequency[0]++;
        } else if (data[i] == 2) {
            frequency[1]++;
        } else {
            frequency[2]++;
        }
    }
    int exchange = 0;
    for (int i = 0; i < frequency[0] + frequency[1]; i++) {
        if (i < frequency[0]) {
            if (data[i] == 2) {
                for (int j = frequency[0]; j < size; j++) {
                    if (data[j] == 1) {
                        std::swap(data[i], data[j]);
                        exchange++;
                        break;
                    }
                }
            } else if (data[i] == 3) {
                for (int j = size - 1; j >= frequency[0]; j--) {
                    if (data[j] == 1) {
                        std::swap(data[i], data[j]);
                        exchange++;
                        break;
                    }
                }
            }
        } else {
            if (data[i] == 3) {
                for (int j = frequency[0] + frequency[1]; j < size; j++) {
                    if (data[j] == 2) {
                        std::swap(data[i], data[j]);
                        exchange++;
                        break;
                    }
                }
            }
```

```
48          }
49      }
50      std::cout << exchange << std::endl;
51      return 0;
52  }
```

3 Консоль

```
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab8/cmake-build-debug$ ./DA_lab8
3
3 2 1
1
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab8/cmake-build-debug$ ./DA_lab8
5
3 1 1 1 2
2
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab8/cmake-build-debug$ ./DA_lab8
6
1 1 2 2 3 3
0
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab8/cmake-build-debug$ ./DA_lab8
7
3 2 3 1 2 1 3
3
```

4 Тест производительности

Тест производительности: из файла читаются образцы и текст соответственно.На вход подается 10000,100000 и 1000000 образцов.
для 10000 образцов:
ilya@LAPTOP-1LJF48LQ:$\backslash mnt\backslash c2course\_lab8 - build - debug$ $\$./benchmark < test.txt$
21
для 100000 образцов:
ilya@LAPTOP-1LJF48LQ:$\backslash mnt\backslash c2course\_lab8 - build - debug$ $\$./benchmark < test.txt$
1928
для 1000000 образцов:
ilya@LAPTOP-1LJF48LQ:$\backslash mnt\backslash c8course\_lab5 - build - debug$ $\$./benchmark < test.txt$
189803

Как видно,наблюдается квадратичная зависимость по времени.

5 Выводы

Выполнив восьмую лабораторную работу по курсу «Дискретный анализ» я познакомился с идеями жадных алгоритмов. Жадные алгоритмы применимы далеко не всегда, в силу того что требуется доказывать их оптимальность, которая как правило доказывется следующим образом: доказывается что оптимальный выбор на первом шаге не закрывает оптимальное решение, после этого показывается, что полученная задача аналогична исходной, кроме того такие задачи как правило основываются на том что оптимальное решение задачи содержит оптимальное решение всех её подзадача. Таким образом, несмотря на то, что подобные алгоритмы могут быть очень эффективными, применимость их весьма ограничена.

Список литературы

[1] *Жадный алгоритм*
URL: `https://ru.wikipedia.org/wiki/Жадный_алгоритм` (дата обращения: 20.03.2021).

[2] *Хабр - Жадные алгоритмы*
URL: `https://habr.com/ru/post/120343` (дата обращения: 20.03.2021).