Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Дискретный анализ»

Студент: Л. А. Фокин
Преподаватель: С. А. Михайлова
Группа: М8О-201Б-23

Дата: 24.03.2025
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1

Задача: Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.

Вариант сортировки: Сортировка подсчётом.

Тип ключа: `Целые числа от 0 до 65535.`

Тип значения: `Целые числа от 0 до` $2^{64} - 1$.

1 Описание

Основная идея сортировки подсчетом заключается в том, чтобы для каждого входного элемента x определить количество элементов, которые меньше x [1].

Алгоритм сортировки подсчётом выглядит следующим образом:

1. Найти максимальный и минимальный элементы max и min в массиве $data$.
2. Создать массив $count$ размером $max - min + 1$ и заполнить его нулями.
3. Посчитать количество вхождений каждого элемента в исходном массиве $data$.
4. Вычислить префиксные суммы в массиве $count$.
5. Создать выходной отсортированный массив $sortedData$, заполнив его с использованием информации из массива $count$ о позициях элементов.

2 Исходный код

Исходный код реализует сортировку подсчётом для пар «ключ-значение», где ключ и значение представляют собой целые числа. Программа считывает входные данные построчно, разделяя ключ и значение по символу табуляции. Затем определяются максимальный и минимальный ключи, которые используются для создания массива count размером $max - min + 1$. После этого программа подсчитывает количество вхождений каждого ключа в массиве count, вычисляет префиксные суммы и создает выходной отсортированный массив sortedData. Наконец, программа выводит отсортированные пары ключ-значение.

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <utility>
#include <cstdio>
#include <ctime>
const int MAX = 65536;

void countSort(std::vector<std::pair<unsigned short, unsigned long long>> &data) {
    if (data.empty()) return;

    unsigned short minKey = std::min_element(data.begin(), data.end(),
        [](const auto &a, const auto &b){ return a.first < b.first; })->first;
    unsigned short maxKey = std::max_element(data.begin(), data.end(),
        [](const auto &a, const auto &b){ return a.first < b.first; })->first;

    std::vector<unsigned int> counts(maxKey - minKey + 1, 0);

    for (const auto &elem : data) {
        counts[elem.first - minKey]++;
    }

    for (size_t i = 1; i < counts.size(); i++) {
        counts[i] += counts[i - 1];
    }

    std::vector<std::pair<unsigned short, unsigned long long>> sortedData(data.size());
    for (int i = data.size() - 1; i >= 0; i--) {
        sortedData[counts[data[i].first - minKey] - 1] = data[i];
        counts[data[i].first - minKey]--;
    }

    data = std::move(sortedData);
}

int main(int argc, char *argv[]) {
    std::vector<std::pair<unsigned short, unsigned long long>> data;
```

```
38      FILE *inFile = fopen(argv[1], "r");
39      unsigned short first;
40      unsigned long long second;
41      while (fscanf(inFile, "%hu\t%llu", &first, &second) == 2) {
42          data.push_back(std::make_pair(first, second));
43      }
44      fclose(inFile);
45
46      clock_t start = clock();
47
48      countSort(data);
49
50      clock_t end = clock();
51      double elapsed = double(end - start) / CLOCKS_PER_SEC;
52
53      std::cout << "Elapsed time: " << elapsed << " seconds" << std::endl;
54
55      FILE *outFile = fopen(argv[2], "w");
56      for (size_t i = 0; i < data.size(); i++) {
57          fprintf(outFile, "%hu\t%llu\n", data[i].first, data[i].second);
58      }
59      fclose(outFile);
60
61
62      return 0;
63  }
```

3 Консоль

```
g++ -o build/main main.cpp
./build/main input.txt output.txt

[input.txt]
0 13207862122685464576
65535 76703883147078533312
0 4588010303972900864
65535 12992997081104908288

[output.txt]
0 13207862122685464576
0 4588010303972900864
65535 76703883147078533312
65535 12992997081104908288
```

4 Тест производительности

1 Условия тестирования

Тестирование производительности проводилось на 100'000'000 строках, с ключами от 0 до 65535 и значениями от 0 до $2^{64} - 1$. Время работы стандартной сортировки `std::stable_sort` составило 47.5887 секунд, а время работы сортировки подсчётом — 6.36868 секунд.

```
discrete-labs git:(lab1) make test ARGS="./data/wide_range.txt output.txt"
Testing std::stable_sort...
Elapsed time: 47.5887 seconds
Testing countSort...
Elapsed time: 6.36868 seconds
```

Сортировка подсчётом выполнилась в 7.46 раз быстрее, чем стандартная сортировка. Это объясняется тем, что сортировка подсчётом `countSort` имеет линейную временную сложность $O(n + k)$, где n — количество элементов, а k — диапазон возможных значений ключей. В данном случае диапазон ключей ограничен (от 0 до 65535), что делает алгоритм особенно эффективным. При этом стандартная сортировка `std::stable_sort`, обладающая временной сложностью $O(n \log n)$, требует больше времени для обработки больших объёмов данных и не учитывает специфику данных. Поэтому, такая разница в скорости является обоснованной и ожидаемой.

5 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я узнал о такой сортировке, как сортировка подсчётом. Она позволяет эффективно сортировать данные, когда диапазон возможных значений ключей ограничен. В ходе работы я реализовал алгоритм сортировки подсчётом для пар «ключ-значение», где ключ и значение представляют собой целые числа. Я протестировал его на большом объёме данных и сравнил с стандартной сортировкой `std::stable_sort`. Результаты показали, что сортировка подсчётом значительно быстрее стандартной сортировки, что подтверждает её эффективность в определённых условиях.

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] *Сортировка подсчётом — Википедия.*
URL: `http://ru.wikipedia.org/wiki/Сортировка_подсчётом`