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: Т. А. Бучкин
Преподаватель: С. А. Михайлова
Группа: М8О-201Б

Дата: 21.02.2024
Оценка:
Подпись:

Москва, 2024

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.

Вариант ключа: `Числа от 0 до 65535.`

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

1 Описание

Требуется написать реализацию алгоритма сортировки подсчётом.

Как сказано в [1]: «основная идея сортировки подсчетом заключается в том, чтобы для каждого входного элемента x определить количество элементов, которые меньше x».

В качестве небольшого улучшения сдвинем диапазон входных числа к нулю слева, вычев из всех чисел минимальное. Это позволит использовать отрицательные числа как индекс в массиве счётчика, а также, если левая граница диапазона больше нуля, сжать размер массива счётчика до размера диапазона, а не максимального числа.

2 Исходный код

```cpp
#include <iostream>

using namespace std;

#ifndef COUNTING_SORT_HPP_
#define COUNTING_SORT_HPP_
/*
    Define class CountingSort that response for sorting
    Define struct Item that will contain key-value pairs
*/
#include <vector>

struct Item {
    unsigned int key;
    unsigned long long value;
};

namespace sorts {

class CountingSort {
private:
    unsigned int minElement_;
    unsigned int maxElement_;

    void checkForMinMaxElements(vector<Item> & items) { // find min and max keys
        minElement_ = items[0].key;
        maxElement_ = items[0].key;
        for (auto & item : items) {
            unsigned int currentKey = item.key;
            if (currentKey < minElement_) {
                minElement_ = currentKey;
            }

            if (currentKey > maxElement_) {
                maxElement_ = currentKey;
            }
        }
    }

public:
    CountingSort() = default;

    void sort(vector<Item> & items) { // sorting
        if (items.size() == 0) {
            return;
        }

```

```
48          checkForMinMaxElements(items);
49          vector<Item> tempItems(items); /*
50              Create items's copy
51              Vector-argument will be sorted
52          */
53          vector<unsigned int> elementsCounter(maxElement_ - minElement_ + 1);
54
55          for (auto & item : items) {
56              elementsCounter[item.key - minElement_]++;
57          }
58
59          for (int i = 1; i < elementsCounter.size(); ++i) {
60              elementsCounter[i] += elementsCounter[i - 1];
61          }
62
63
64          for (int i = tempItems.size() - 1; i >= 0; --i) {
65              long index = --elementsCounter[tempItems[i].key - minElement_];
66              items[index] = tempItems[i];
67          }
68      }
69  };
70
71  } // !sorts
72
73  #endif // !COUNTING_SORT_HPP_
74
75
76  int main() {
77      ios::sync_with_stdio(false); // Setting that busts IOput
78      cin.tie(0);
79
80      vector<Item> data;
81
82      unsigned int key;
83      unsigned long long value;
84      while (cin >> key >> value) {
85          data.emplace_back(Item{ key, value });
86      }
87
88      sorts::CountingSort sorter;
89      sorter.sort(data);
90
91      for (auto & item : data) {
92          cout << item.key << '\t' << item.value << '\n'; // Using endl will call flush
                buffer so i use '\n'
93      }
94
95      return 0;
```

```
96 || }
```

3 Консоль

```
PS C:/Users/User/Desktop/Learning/2Course/Discrete-Analysis>cmake --build ./build
--config Debug --target main -j 10 --
PS C:/Users/User/Desktop/Learning/2Course/Discrete-Analysis>cd build
PS C:/Users/User/Desktop/Learning/2Course/Discrete-Analysis/build>./main.exe
0 13207862122685464576
65535 7670388314707853312
0 4588010303972900864
65535 12992997081104908288
^D
0 13207862122685464576
0 4588010303972900864
65535 7670388314707853312
65535 12992997081104908288
```

4 Тест производительности

В сравнении с стабильной сортировкой из стандартной библиотеки STL сортировка подсчётом показала много худший результат на экстра малых объёмах данный(порядка 10 элементов), в силу затрат на дополнительные массивы.

На больших объёмах данных (порядка 1 000 000 элементов) сортировка подсчётом показала кратно лучший результат, что доказывает её линейную асимптотику.

```
raison@WIN-4SUTO50B1V5:/mnt/c/Users/User/Desktop/Learning/2Course/
Discrete-Analysis/Lab␣1$␣./benchmark␣<␣./tests/01.t
Count␣of␣lines␣is␣0
Counting␣sort␣time:␣0us
STL␣stable␣sort␣time:␣0us

raison@WIN-4SUTO50B1V5:/mnt/c/Users/User/Desktop/Learning/2Course/
Discrete-Analysis/Lab␣1$␣./benchmark␣<␣./tests/02.t
Count␣of␣lines␣is␣1
Counting␣sort␣time:␣6us
STL␣stable␣sort␣time:␣2us

raison@WIN-4SUTO50B1V5:/mnt/c/Users/User/Desktop/Learning/2Course/
Discrete-Analysis/Lab␣1$␣./benchmark␣<␣./tests/03.t
Count␣of␣lines␣is␣10
Counting␣sort␣time:␣1097us
STL␣stable␣sort␣time:␣6us

raison@WIN-4SUTO50B1V5:/mnt/c/Users/User/Desktop/Learning/2Course/
Discrete-Analysis/Lab␣1$␣./benchmark␣<␣./tests/08.t
Count␣of␣lines␣is␣1000000
Counting␣sort␣time:␣112591us
STL␣stable␣sort␣time:␣721206us
```

5 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился реализовывать сортировку подсчётом, поразрядную и карманную сортировки.

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.))