

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: Е. С. Пищик
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б
Дата:
Оценка:
Подпись:

Москва, 2020

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: Поразрядная сортировка.

Вариант ключа: Автомобильные номера в формате А 999 ВС (используются буквы латинского алфавита).

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

1 Описание

Требуется написать реализацию алгоритма поразрядной сортировки. Основная идея такой сортировки заключается в том, чтобы разделить ключ на разряды, для каждого разряда запустить сортировку подсчётом, начиная с самого правого разряда, количество разрядов у каждого элемента x должно быть одинаковое.

Как сказано в [1]: «Сортировка подсчётом — алгоритм сортировки, в котором используется диапазон чисел сортируемого массива для подсчёта совпадающих элементов. Применение сортировки подсчётом целесообразно лишь тогда, когда сортируемые числа имеют диапазон возможных значений, который достаточно мал по сравнению с сортируемым множеством, например, миллион натуральных чисел меньших 1000».

2 Исходный код

main.cpp	
int main()	Функция, использующая все написанные функции из других файлов, для создание вектора, заполнения вектора, сортировки вектора и печати вектора.
structs.hpp	
struct TPair{}	Структура, хранящая пару «ключ-значение».
my_vector.hpp	
class TVector{}	Класс собственного динамического массива.
radix_sort.hpp	
size_t TmpChange(TVector<T>& vec, size_t count, size_t i)	Функция возвращающая различные значения, возвращаемое значение зависит от значения count
void CountingSort(TVector<T>& vec, size_t const& max, size_t& count)	Функция сортировки подсчётом.
void RadixSort(TVector<T>& vec, size_t const& max_1, size_t const& max_2, size_t const& max_3)	Функция поразрядной сортировки.

```

1  typedef unsigned long long TULL;
2  struct TPair
3  {
4      TPair(){}
5      TPair(std::string const& key_0, std::string const& key_1, std::string const& key_2,
6            TULL const& val);
7      TPair(TPair const& obj);
8      std::string key_[3];
9      TULL val_;
10 };
11 template<typename T>
12 class TVector
13 {
14     private:
15         size_t size_;
16         size_t cap_;
17         T* data_;
18     public:
19         TVector();
20         TVector(size_t cap);
21         ~TVector();

```

```
21 | TVector(TVector<T> const& obj);
22 | T& operator[](size_t index) const;
23 | T& operator[](size_t index);
24 | void PushBack(T const& val);
25 | void Print() const;
26 | size_t Size() const;
27 |};
```

3 Консоль

```
pe4eniks@pe4eniks-HP-Laptop-14-dk0xxx:~/solution/solution$ make
g++ -Wall -std=c++11 -O2 -lm -Werror -Wno-sign-compare -pedantic -c -o main.o
main.cpp
g++ -Wall -std=c++11 -O2 -lm -Werror -Wno-sign-compare -pedantic -o solution
main.cpp
pe4eniks@pe4eniks-HP-Laptop-14-dk0xxx:~/solution/solution$ cat test.txt
A 999 AA 6315351
Z 000 ZZ 5454
A 999 AA 12
Z 000 ZZ 123455666
pe4eniks@pe4eniks-HP-Laptop-14-dk0xxx:~/solution/solution$ ./solution <test.txt
A 999 AA 6315351
A 999 AA 12
Z 000 ZZ 5454
Z 000 ZZ 123455666
```

4 Тест производительности

Тест производительности представляет из себя следующее: в файле лежит 1 миллион строк с парами «ключ-значение», которые сортируются поразрядной сортировкой и `std::stable_sort`.

```
da_exercise_01.exe < test.txt
Count of lines is 1000000
Radix sort time: 7440ms
STL stable sort time: 16440ms
```

Как видно, время работы поразрядной сортировки значительно меньше, чем время работы `stable_sort`.

5 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился оценивать сложность алгоритмов, работать с шаблонами, сложными структурами данных, разделять проект на различные файлы, работать с несколькими файлами в проекте, изучил сортировку подсчётом и поразрядную сортировку.

Список литературы

[1] *Сортировка подсчётом* — Википедия.

URL: https://ru.wikipedia.org/wiki/Сортировка_подсчётом