

**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: А. Т. Бахарев
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б
Дата:
Оценка:
Подпись:

Москва, 2017

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: MD5-суммы (32-разрядные шестнадцатичные числа).

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

1 Описание

Требуется написать реализацию алгоритма поразрядной сортировки.

Идея сортировки заключается в разбиении сортируемых элементов на разряды. Затем выполняется устойчивая сортировка подсчетом для каждого разряда. При этом, для строк подходит версия MSD(Most Significant Digit) - сортировка начинается от самого старшего разряда. Для чисел же нужно использовать LSD-версию(Least Significant Digit). В данной версии, сортировка начинается от самого младшего разряда.

2 Исходный код

На каждой непустой строке входного файла располагается пара «ключ-значение». Ключ представляет собой строку из 32 элементов. Значение – целое неотрицательное число. Для хранения такой пары создается динамический массив *array* из объектов класса *Telement*. Внутри этого класса содержатся непосредственно сама пара, а также 2 указателя на ключ и значение. Данная особенность класса необходима для корректной сортировки строк без их изменения. Будут меняться только ссылки на соответствующие ключи и значения.

В случае, если код не помещается на одну-две страницы *A4*, тогда следует сделать табличку следующего вида:

main.c	
void sort(struct KV & B, struct KV & Res, int max, int size)	Функция сортировки подсчётом
file1.c	
void function_name()	Функция, «которая почти всегда работает, но неясно, что она делает».

В этом случае структуры или классы должны быть полностью приведены в листинге (без реализации методов).

```
1 | struct KV{
2 |     int key;
3 |     char value;
4 | } KV;
```

3 Консоль

```
a.kukhticev$ gcc -pedantic -Wall -std=c99 -Werror -Wno-sign-compare -lm da10.c
-o da10 --some_long_argument=true
a.kukhticev$ cat test1
87 a
13 b
89 c
13 d
a.kukhticev$ ./da10 <test1
13 b
13 d
87 a
89 c
```

4 Тест производительности

Тут Вы описываете собственно тест производительности, сравнение Вашей реализации с уже существующими и т.д.

Тест производительности представляет из себя следующее: поиск образцов с помощью суффиксного массива сравнивается с поиском алгоритма КМП, но время на построение суффиксного массива не учитывается. Текст состоит из 1 миллиона букв: а образцов около 200 штук, длина которых может быть от 2 до 100 букв.

```
Andys-MacBook-Pro:kmp Andy$ g++ main.cpp
Andys-MacBook-Pro:kmp Andy$ ./a.out <../in.txt >out2.txt
Andys-MacBook-Pro:kmp Andy$ cat out2.txt | grep "time"
KMP search time: 1.639993 sec
Andys-MacBook-Pro:sa Andy$ make
g++ -pedantic -Wall -Werror -Wno-sign-compare -Wno-long-long -lm -O2 -o lab5
main.cpp suffix_tree.cpp suffix_array.cpp
Andys-MacBook-Pro:sa Andy$ ./lab5 <../in.txt >out1.txt
Andys-MacBook-Pro:sa Andy$ cat out1.txt | grep "time"
Suffix array build time: 2.179744 sec
Suffix array search time: 0.003511 sec
```

Как видно, что суффиксный массив выиграл у КМП, так как и т.д.

5 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился реализовывать и применять на практике эффективные сортировки за линейное время. Те алгоритмы, которые я знал до этого, значительно проигрывают по времени и по памяти. Также, я реализовал свой класс для строк, узнав при этом о внутреннем устройстве этого контейнера. Я научился писать бенчмарки для оценки времени работы программы, а также определять сложность работы алгоритмов.

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 (дата обращения: 16.12.2013).
- [3] Список использованных источников оформлять нужно по ГОСТ Р 7.05-2008