

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: В. И. Пупкин
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: Сортировка подсчётом.

Вариант ключа: Числа от 0 до 65535.

Вариант значения: Строки фиксированной длины 64 символа, во входных данных могут встретиться строки меньшей длины, при этом строка дополняется до 64-х нулевыми символами, которые не выводятся на экран.

1 Описание

Требуется написать реализацию алгоритма сортировки подсчётом. Турум-пум-пум... Как сказано в [1]: «основная идея сортировки подсчетом заключается в том, чтобы для каждого входного элемента x определить количество элементов, которые меньше x ».

Или можно так:

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

2 Исходный код

Здесь должно быть подробное описание программы и основные этапы написания кода.

На каждой непустой строке входного файла располагается пара «ключ-значение», поэтому создадим новую структуру *KV*, в которой будем хранить ключ и значение. И так далее.

```
1 #include "stdio.h"
2 #include "stdlib.h"
3
4 struct KV{
5     int key;
6     char value;
7 } KV;
8
9 void sort(struct KV * B, struct KV * Res, int max, int size){
10     int C[max+1];
11     for(int i = 0; i <= max; i++)
12         C[i] = 0;
13     for(int i = 0; i < size; i++){
14         int idx = B[i].key;
15         C[idx]++;
16     }
17     for(int i = 1; i <= max; i++)
18         C[i] = C[i] + C[i-1];
19
20     for(int i = size-1; i >= 0; i--){
21         int idx = B[i].key;
22         int pos = C[idx];
23         Res[pos-1] = B[i];
24         C[idx]--;
25     }
26 }
27
28 int main()
29 {
30     int key;
31     char value;
32     struct KV * A = NULL;
33     int size = 0;
34     int max = 0;
35
36     while (scanf("%d\t%c",&key, &value) != EOF)
37     {
38         size++;
39         A = (struct KV *)realloc(A, size*sizeof(struct KV));
40         A[size-1].key = key;
```

```

41 |     A[size-1].value = value;
42 |     if(key > max)
43 |         max = key;
44 | }
45 |
46 | struct KV Res[size];
47 | sort(A, Res, max, size);
48 | for(int i = 0; i < size; i++)
49 |     printf("%d\t%c\n", Res[i].key, Res[i].value);
50 |
51 |     free(A);
52 |
53 | }

```

В случае, если код не помещается на одну-две страницы А4, 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

Здесь Вы пишете то, чему научились на лабораторной на самом деле, что узнали нового, где это может пригодиться и т.д. Мне важно, какие именно Вы сделали выводы из лабораторной.

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился тому-то и тому-то.

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