

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: Г. А. Ермаков
Преподаватель: С. А. Михайлова
Группа: М8О-201Б
Дата:
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: Поразрядная сортировка.

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

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

Основная идея алгоритма решения задачи состоит в применении поразрядной сортировки, которая позволяет упорядочивать автомобильные номера в формате А 999 ВС за линейное время благодаря их структурированности. Для реализации необходимо обработать ключи по разрядам в порядке от младшего к старшему, начиная с цифр, затем букв, при этом на каждом этапе применяется сортировка подсчетом для текущего разряда. Сортировка подсчетом идеально подходит, поскольку разряды ключей ограничены фиксированным количеством значений (10 для цифр и 26 для каждой из буквенных групп). Тип значения, представляющий строки фиксированной длины до 64 символов, требует предварительной нормализации путем добавления нулевых символов, чтобы все строки стали одинаковой длины. После нормализации строки участвуют в сортировке как привязанные к ключам данные и выводятся в соответствии с упорядоченными ключами, при этом нулевые символы не отображаются в результате.

2 Исходный код

На каждой непустой строке входного файла располагается пара «ключ-значение», в которой ключ указан согласно заданию, затем следует знак табуляции и указано соответствующее значение.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  struct Entry {
5      array<uint8_t, 6> key;
6      char value[65];
7  };
8
9  int main() {
10     ios::sync_with_stdio(false);
11     cin.tie(nullptr);
12
13     vector<Entry> entries;
14     string line;
15
16     while (getline(cin, line)) {
17         if (line.empty()) continue;
18         auto tab = line.find('\t');
19
20         string key_s;
21         key_s.reserve(6);
22         for (char c : line.substr(0, tab)) {
23             if (isalnum(static_cast<unsigned char>(c)))
24                 key_s.push_back(c);
25         }
26         if (key_s.size() != 6) continue;
27
28         Entry e;
29
30         auto val = line.substr(tab + 1);
31         size_t len = min(val.size(), size_t(64));
32         memcpy(e.value, val.data(), len);
33         e.value[len] = '\0';
34
35         for (int i = 0; i < 6; ++i) {
36             char c = key_s[i];
37             e.key[i] = isalpha(static_cast<unsigned char>(c)) ? c - 'A' : c - '0';
38         }
39         entries.push_back(e);
40     }
41
42     size_t n = entries.size();
43     vector<Entry> buffer(n);
```

```

44 | array<int, 26> cnt;
45 | array<int, 26> pos_arr;
46 |
47 | for (int d = 5; d >= 0; --d) {
48 |     int range = (d == 0 || d >= 4 ? 26 : 10);
49 |     cnt.fill(0);
50 |     for (auto &e : entries) cnt[e.key[d]]++;
51 |     pos_arr[0] = 0;
52 |     for (int i = 1; i < range; ++i)
53 |         pos_arr[i] = pos_arr[i - 1] + cnt[i - 1];
54 |
55 |     for (auto &e : entries) {
56 |         int k = e.key[d];
57 |         buffer[pos_arr[k]++] = e;
58 |     }
59 |     entries.swap(buffer);
60 | }
61 |
62 | for (auto &e : entries) {
63 |     cout << char('A' + e.key[0]) << ' ';
64 |     cout << int(e.key[1]) << int(e.key[2]) << int(e.key[3]) << ' ';
65 |     cout << char('A' + e.key[4]) << char('A' + e.key[5]);
66 |     cout << '\t' << e.value << '\n';
67 | }
68 | return 0;
69 | }

```

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