

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: Меджидли И. И. о  
Преподаватель: Михайлова С. А  
Группа: М8О-201Б-21  
Дата: 28.03.2023  
Оценка:  
Подпись:

Москва, 2023

## 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 переменной длины (до 2048 символов).

# 1 Описание

Требуется написать реализацию алгоритма сортировки подсчётом.

Идея:

Проходимся по вектору пар <ключ-строка> (входные данные), который был создан до сортировки по исходным данным, находим максимальное значение ключа (далее буду называть  $n$ ). Создаем вектор чисел (ключей) с количеством элементов, равным  $n$ . Пройдя еще раз по первому вектору, в векторе ключей для каждого ключа заполняем сколько раз он встречался в векторе пар (первом векторе). В векторе ключей к значению в каждой ячейке, начиная с первой (с индексом 1, не 0), прибавляется значение предыдущей ячейки. После этого создается еще один вектор пар <ключ-строка> для отсортированных пар такого же размера, как и первый вектор. Проходя по первому вектору справа налево, для каждой пары находится нужная ячейка в третьем векторе следующим образом: при обратном прохождении у каждой пары первого вектора берется ключ, по нему находится количество пар находящихся до текущей пары, включая эту (далее -  $k$ ), - значение ячейки во втором векторе (лежит на месте значения ключа). После этого из значения ячейки вычитается 1. Наконец, на место  $k - 1$  ставится пара <ключ-строка>. И так далее...

После заполнения третьего вектора он свапается с первым, чтобы было удобнее выводить результат в программе.

## 2 Исходный код

Основные этапы:

- 1)Создания вектора пар ключ-строка.
- 2)Заполнения вектора исходными данными.
- 3)Выполнение функции сортировки подсчетом.
- 4)Вывод получившегося вектора.

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <algorithm>
5  using namespace std;
6
7
8  void counting_sort( vector<pair<int, string>> &elems )
9  {
10     if (elems.empty())
11     {
12         return ;
13     }
14
15     int32_t max_elem = elems[0].first;
16
17     for (size_t i = 1; i < elems.size(); ++i)
18     {
19         auto pair = elems[i];
20         max_elem = max(max_elem, pair.first);
21     }
22
23     vector<int> tmp(max_elem + 1);
24
25     for (size_t i = 0; i < elems.size(); ++i)
26     {
27         ++tmp[elems[i].first];
28     }
29
30     for (size_t i = 1; i < tmp.size(); ++i)
31     {
32         tmp[i] += tmp[i-1];
33     }
34
35     vector<pair<int, string>> result( elems.size() );
36     for (int i = elems.size()-1; i >= 0; --i )
37     {
38         auto key = elems[i].first;
39         auto pos = tmp[key]--;
40         result[pos-1] = elems[i];
```

```

41     }
42
43     swap(elems, result);
44 }
45
46
47 int main()
48 {
49     ios::sync_with_stdio(0); cin.tie(0);
50     int key;
51     string value;
52
53     vector<pair<int, string>> elems;
54     while (cin >> key >> value)
55     {
56         elems.push_back( make_pair(key, value) );
57     }
58
59     // Counting sort.
60     counting_sort( elems );
61
62     for (auto& elem : elems)
63     {
64         cout << elem.first << "\t" << elem.second << "\n";
65     }
66
67     return 0;
68 }

```

### 3 Тесты

Тестировать программу буду с помощью подачи текстового файла с нужной структурой (пара значений ключ-строка)

```
imedzhidli@imedzhidli:~/Desktop/DA$ cat test
0 xGfxrxGGxrxMMMMfrrrG
65535 xGfxrxGGxrxMMMMfrrr
0 xGfxrxGGxrxMMMMfrr
65535 xGfxrxGGxrxMMMMfrr
5 xGfxsarxGGxrxtMMfr
imedzhidli@imedzhidli:~/Desktop/DA$ ./laba1 <test
0 xGfxrxGGxrxMMMMfrrrG
0 xGfxrxGGxrxMMMMfrr
5 xGfxsarxGGxrxtMMfr
65535 xGfxrxGGxrxMMMMfrrr
65535 xGfxrxGGxrxMMMMfrr
imedzhidli@imedzhidli:~/Desktop/DA$
```

Как можно заметить, все верно. А теперь сделаем стресс-тест нашей программы, отправив 100 строк:

```
imedzhidli@imedzhidli:~/Desktop/DA$ cat test2
0 xGfxrxGGxrxMMMMfrrrG
65535 xGfxrxGGxrxMMMMfrrr
0 xGfxrxGGxrxMMMMfrr
65535 xGfxrxGGxrxMMMMfrr
5 xGfxsarxGGxrxtMMfr
10 xGfxtxGGxrxMMMtfr
1 xGfxrxGGxrxMMMMfrrrM
2 xGfxpxGGxrxMMMMfrrrN
50 xGfxrxGGxrxMMMMfrrrP
0 xGfxrxGGxrxMMMMfrrrQ
65535 xGfxrxGGxrxMMMMfrrrR
3 xGfxqxGGxrxMMMMfrrrT
25 xGfxrxGGxrxMMMMfrrrU
0 xGfxrxGGxrxMMMMfrrrV
65535 xGfxrxGGxrxMMMMfrrrW
4 xGfxwxGGxrxMMMMfrrrX
12 xGfxrxGGxrxMMMMfrrrY
```

0 xGfxrxGGxrxMMMMfrrrZ  
 65535 xGfxrxGGxrxMMMMfrrrA  
 6 xGfxaxGGxrxMMMMfrrrB  
 18 xGfxrxGGxrxMMMMfrrrC  
 0 xGfxrxGGxrxMMMMfrrrD  
 65535 xGfxrxGGxrxMMMMfrrrE  
 7 xGfxexGGxrxMMMMfrrrF  
 36 xGfxrxGGxrxMMMMfrrrH  
 0 xGfxrxGGxrxMMMMfrrrI  
 65535 xGfxrxGGxrxMMMMfrrrJ  
 8 xGfxjxGGxrxMMMMfrrrK  
 9 xfmxxnGGxrxMMfrN  
 0 xfnopxxrxMMfrQ  
 65535 xfqrxxxrxMMfrR  
 11 xfSxtxxrxMMfrT  
 13 xfUxxvxxrxMMfrV  
 0 xfWxxwxxrxMMfrX  
 65535 xfYxxzxxrxMMfrZ  
 14 xfAxxaxrxMMfrB  
 15 xfCxxbxxrxMMfrD  
 0 xfExxcxxrxMMfrF  
 65535 xfHxxdxxrxMMfrH  
 16 xfIxxexrxMMfrJ  
 17 xfKxxgxxrxMMfrL  
 0 xfMxxhxxrxMMfrN  
 65535 xfOxxixrxMMfrP  
 18 xfQxxjxxrxMMfrR  
 19 xfTxxkxxrxMMfrT  
 0 xfVxxlxxrxMMfrV  
 65535 xfXxmxxxrxMMfrX  
 20 xfZxxnxxrttMFrZ  
 21 xfAxxoxxrttMFrB  
 0 xfCxxpxxrttMFrD  
 65535 xfExrxxxrttMFrF  
 22 xfHxxsxxrttMFrH  
 23 xfIxxtxxrttMFrI  
 0 xfKxxuxxrttMFrL  
 65535 xfMxxvxxrttMFro  
 24 xfOxxwxxrttMFrP  
 25 xfQxxxXXXrttMFrQ  
 0 xfTxxyXXXrttMFrs

65535 xfVxzXXXrttMFru  
26 xfXxaXXXrttMFrw  
27 xfZxbXXXrttMFrz  
0 xfAxcXXXrttMFrA  
65535 xfCxdXXXrttMFrC  
28 xfExeXXXrttMFrE  
29 xfHxfXXXrttMFrH  
30 xfIxfXXXrttMFrI  
31 xfJxfXXXrttMFrJ  
2342 xfKxfXXXrttMFrK  
3453 xfLxfXXXrttMFrL  
324 xfMxfXXXrttMFrM  
5 xfNxfXXXrttMFrN  
236 xfOxfXXXrttMFrO  
17 xfPxfXXXrttMFrP  
0 xfQxfXXXrttMFrQ  
39 xfRxfXXXrttMFrR  
0 xfSxfXXXrttMFrS  
41 xfTxf  
0 xfTxyyXXXrttMFrs  
65535 xfVxzXXXrttMFru  
1 xfXxaXXXrttMFrw  
65534 xfZxbXXXrttMFrz  
2 xfAxcXXXrttMFrA  
65533 xfCxdXXXrttMFrC  
3 xfExeXXXrttMFrE  
65532 xfHxfXXXrttMFrH  
4 xfIxfXXXrttMFrI  
65531 xfJxfXXXrttMFrJ  
5 xfKxfXXXrttMFrK  
65530 xfLxfXXXrttMFrL  
6 xfMxfXXXrttMFrM  
65529 xfNxfXXXrttMFrN  
7 xfOxfXXXrttMFrO  
65528 xfPxfXXXrttMFrP  
8 xfQxfXXXrttMFrQ  
65527 xfRxfXXXrttMFrR  
9 xfSxfXXXrttMFrS  
65526 xfTxfXXXrttMFrT  
10 xfUxfXXXrttMFrU  
65525 xfVxf



```

1918 xfxXxXxXxXxXxXxXx
imedzhidli@imedzhidli:~/Desktop/DA$ ./labal <test2
0 xGfxrxGGxrxMMMMfrrrrG
0 xGfxrxGGxrxMMMMfrr
0 xGfxrxGGxrxMMMMfrrrrQ
0 xGfxrxGGxrxMMMMfrrrrV
0 xGfxrxGGxrxMMMMfrrrrZ
0 xGfxrxGGxrxMMMMfrrrrD
0 xGfxrxGGxrxMMMMfrrrrI
0 xfnopxxrxxtMMfrQ
0 xfWxxwxxrxxtMMfrX
0 xfExxcxxrxxtMMfrF
0 xfMxxhxxrxxtMMfrN
0 xfVxxlxxrxxtMMfrV
0 xfCxxpxxrttMFrD
0 xfKxxuxxxrttMFrI
0 xfTxxyXXXrttMFrs
0 xfAxcXXXrttMFrA
0 xfQxfXXXrttMFrQ
0 xfSxfXXXrttMFrS
0 xfTxxyXXXrttMFrs
1 xGfxrxGGxrxMMMMfrrrrM
1 xfXxaXXXrttMFrw
2 xGfxpxGGxrxMMMMfrrrrN
2 xfAxcXXXrttMFrA
3 xGfxqxGGxrxMMMMfrrrrT
3 xfExeXXXrttMFrE
4 xGfxwxGGxrxMMMMfrrrrX
4 xflxfXXXrttMFrI
5 xGfxsarxGGxrxxtMMfr
5 xfNxfXXXrttMFrN
5 xfKxfXXXrttMFrK
6 xGfxaxGGxrxMMMMfrrrrB
6 xfMxfXXXrttMFrM
7 xGfxexGGxrxMMMMfrrrrF
7 xfOxfXXXrttMFrO
8 xGfxjxGGxrxMMMMfrrrrK
8 xfQxfXXXrttMFrQ
9 xfmxxnxGGxrxxtMMfrN
9 xfSxfXXXrttMFrS
10 xGfxtxGGxrxMMMtfr

```

10 xfUxfXXXrttMFrU  
 11 xfSxtxxxxrxtMMfrT  
 12 xGfxrxGGxrxMMMMfrrrY  
 13 xfUxxvxxxrxtMMfrV  
 14 xfAxxaxxrxtMMfrB  
 15 xfCxxbxxxrxtMMfrD  
 16 xfLxxexxrxtMMfrJ  
 17 xfKxxgxxxrxtMMfrL  
 17 xfPxfXXXrttMFrP  
 18 xGfxrxGGxrxMMMMfrrrC  
 18 xfQxxjxxxrxtMMfrR  
 19 xfTxxkxxxrxtMMfrT  
 20 xfZxxnxxxrttMFrZ  
 21 xfAxxoxxrxtMFrB  
 22 xfHxxsxxxrttMFrh  
 23 xfLxxtxxxrttMFrI  
 24 xfOxxwxxxrttMFrP  
 25 xGfxrxGGxrxMMMMfrrrU  
 25 xfQxxxXXXrttMFrq  
 26 xfXxaXXXrttMFrw  
 27 xfZxbXXXrttMFrz  
 28 xfExeXXXrttMFrE  
 29 xfHxfXXXrttMFrH  
 30 xfLxfXXXrttMFrI  
 31 xfJxfXXXrttMFrJ  
 36 xGfxrxGGxrxMMMMfrrrH  
 39 xfRxfXXXrttMFrR  
 41 xfTxf  
 50 xGfxrxGGxrxMMMMfrrrP  
 236 xfOxfXXXrttMFrO  
 324 xfMxfXXXrttMFrM  
 1918 xfxXxXxXxXxXxXxXx  
 2342 xfKxfXXXrttMFrK  
 3453 xfLxfXXXrttMFrL  
 65525 xfVxf  
 65526 xfTxfXXXrttMFrT  
 65527 xfRxfXXXrttMFrR  
 65528 xfPxfXXXrttMFrP  
 65529 xfNxfXXXrttMFrN  
 65530 xfLxfXXXrttMFrL  
 65531 xfJxfXXXrttMFrJ

```
65532 xfHxfXXXrttMFrH
65533 xfCxdXXXrttMFrC
65534 xfZxbXXXrttMFrz
65535 xGfxrxGGxrxMMMMfrrr
65535 xGfxrxGGxrxMMMMfrr
65535 xGfxrxGGxrxMMMMfrrrrR
65535 xGfxrxGGxrxMMMMfrrrrW
65535 xGfxrxGGxrxMMMMfrrrrA
65535 xGfxrxGGxrxMMMMfrrrrE
65535 xGfxrxGGxrxMMMMfrrrrJ
65535 xfqxrxxxxrxtMMfrR
65535 xfYxxzxxxxrxtMMfrZ
65535 xfHxxdxxxxrxtMMfrH
65535 xfOxxixxrxtMMfrP
65535 xfXxmmxxxxrxtMMfrX
65535 xfExrxxxxrttMFrF
65535 xfMxxvxxxrttMFro
65535 xfVxzXXXrttMFru
65535 xfCxdXXXrttMFrC
65535 xfVxzXXXrttMFru
imedzhidli@imedzhidli:~/Desktop/DA$
```

Опять все верно. Можно сделать вывод, что программа работает исправно.

## 4 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я научился сортировке подсчетом, вспомнил, как использовать структуру `pair`. Я думаю, что знания, полученные в ходе выполнения лабораторной работы, точно понадобятся мне в будущем.