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: М. А. Бронников
Преподаватель: А. А. Кухтичев
Группа: М8О-207Б

Дата:
Оценка:
Подпись:

Москва, 2019

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: `Почтовые индексы`.

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

1 Описание

Как сказано в [1]: «основная идея сортировки подсчетом заключается в том, чтобы для каждого входного элемента x определить количество элементов, которые меньше x».
Число элементов меньше x будет обозначать место элемента в выходном отсортированном массиве.

К особенностям алгоритма относится то, что необходимо определить верхнюю границу k допустимых значений ключей элементов, входящих во входной массив. Если $k = O(n) \Rightarrow k = \Theta(n)$. При этом является стабильной сортировкой и ни одна пара элементов при этом не сравнивается друг с другом.
К минусам можно отнести то, что требует памяти для 2 дополнительных массивов(помимо исходного массива) размерности k и n соответственно.

Алгоритм:

Counting Sort(A):
for $i \leftarrow 0$ **to** k
  $C[i] \leftarrow 0$
for $j \leftarrow 1$ **to** $length[A]$
  $C[A[j]] \leftarrow C[A[j]] + 1$
for $i \leftarrow 1$ **to** k
  $C[i] \leftarrow C[i] + C[i-1]$
for $j \leftarrow length[A]$ **downto** 1
  $B[C[A[j]]] \leftarrow A[j]$
  $C[A[j]] \leftarrow C[A[j]] - 1$

2 Исходный код

На каждой непустой строке входного файла располагается пара «ключ-значение», поэтому создадим новую структуру $Item$, в которой будем хранить ключ и значение каждого элемента структуры $Vector$, которая реализует вектор значений посредством хранения ссылки на первый элемент массива $data$, количества элементов в массиве $buzy$ и максимально возможного количества элементов массива без дополнительного перевыделения памяти len.
Также заранее объявим значения $SUPR$ - верхнюю грань допустимых значений key и $STARTVALUE$ - число, характеризующее выделение и экспоненциальное перевыделение памяти при расширении массива.

```
1  #include <ctype.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5
6  #define STARTVALUE 2
7  #define SUPR 1000000
8
9  typedef unsigned long long int TypeV;
10
11 typedef struct Item{
12   TypeV value;
13   int key;
14 } Item;
15
16 typedef struct vector{
17   unsigned int len;
18   unsigned int buzy;
19   Item* data;
20 } Vector;
21
22 void Count_Sort(Vector* A);
23 void Create_Vector(Vector* A);
24 void Delete_Vector(Vector* A);
25 void Insert_Vector(Vector* A, Item* item);
26 void Read_Item(Vector* A);
27 void Vector_Print(Vector* A);
```

Описание методов и функций:

main.c	
$void\ Count\_Sort(Vector*\ A)$	Метод сортировки подсчётом
$void\ Create\_Vector(Vector*\ A)$	Метод инициализации массива при создании
$void\ Delete\_Vector(Vector*\ A)$	Метод уничтожения массива
$void\ Insert\_Vector(Vector*\ A, Item* item)$	Метод добавления в вектор новой пары ключ-значение
$void\ Read\_Item(Vector*\ A)$	Метод считывания элементов из потока
$void\ Vector\_Print(Vector*\ A)$	Метод печати вектора на экран
$int\ main()$	Основная функция работы

Помимо этого у нас имеется файл *test.c*, в котором реализован генератор тестов для программы, принимающий на вход имя тестового файла и количество тестовых строк.

3 Консоль

```
max@max-X550CC:~/DA/lab1$ gcc -std=c99 -g -Wall main.c -o prog
max@max-X550CC:~/DA/lab1$ gcc -std=c99 test.c -o test.out
max@max-X550CC:~/DA/lab1$ ./test.out
Enter filename of test: test
Enter number of lines: 11
max@max-X550CC:~/DA/lab1$ cat test
334106 677032765
720128 1142281298
980030 627402468
346629 1371614151
128779 1862864535
771972 1008172827
556245 2111525774
446825 759331291
767510 1128162459
044333 245004806
708149 840296678
max@max-X550CC:~/DA/lab1$ ./prog <test
044333 245004806
128779 1862864535
334106 677032765
346629 1371614151
446825 759331291
556245 2111525774
708149 840296678
720128 1142281298
767510 1128162459
771972 1008172827
980030 627402468
```

4 Выводы

Выполнив первую лабораторную работу по курсу «Дискретный анализ», я получил полезные знания о том, как реализуется одна из сортировок за линейное время - сортировка подсчетом, и приобрел свой первый опыт разработки в условиях ограниченной памяти и времени выполения.
Также при выполнении работы я лучше понял и усвоил тему лабораторной рабты, еще раз убедившись в том, что практитика не отделима от теории, ведь только при непосредственной реализации сталкиваешься с неприятными моментами, которые остаются незамеченными при изучении теоретического аспекта, требующими неоднозначных оригинальных и, желательно, изящных решений.

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).