**Федеральное агентство образования**

**Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»**

Факультет компьютерных технологий и информатики

Кафедра МО ЭВМ

Дисциплина: «Построение и анализ алгоритмов.»

**Отчет по лабораторной работе № 1**

Выполнил: студент группы 1381

Грейнер Андрей

Проверил: \_\_\_\_\_\_\_\_\_\_\_

**Санкт-Петербург**

**2013**

**Ход работы:**

1. **Содержательная постановка задачи**

Реализовать и протестировать алгоритм поразрядной сортировки.

1. **Анализ и пример решения задачи.**
   1. ***Анализ задачи***

Необходимо написать функцию, которая будет использовать алгоритм поразрядной сортировки для сортировки списка. А также тестирующую программу, для проверки алгоритма, а так же получения информации о времени его выполнения.

* 1. ***Особенности алгоритм***

1. он совсем не использует сравнений сортируемых элементов.
2. ключ, по которому происходит сортировка, необходимо разделить на части, *разряды* ключа. Например, слово можно разделить по буквам, число - по цифрам...

3.До сортировки необходимо знать два параметра: k и m, где

* k - количество разрядов в самом длинном ключе
* m - разрядность данных: количество возможных значений разряда ключа
  1. ***Пример выполнения программы***

Предположим, что элементы линейного списка L есть k-разрядные десятичные числа, разрядность максимального числа известна заранее. Обозначим d(j,n) - j-ю справа цифра числа n, которую можно выразить как

d(j,n) = [ n / 10j-1 ] % 10

Пусть L0, L1,..., L9 - вспомогательные списки (карманы), вначале пустые. Поразрядная сортировка состоит из двух процессов, называемых распределение и сборка и выполняемых для j=1,2,...,k.

Фаза распределения разносит элементы L по карманам: элементы li списка L последовательно добавляются в списки Lm, где m = d(j, li). Таким образом получаем десять списков, в каждом из которых j-тые разряды чисел одинаковы и равны m.

Фаза сборки состоит в объединении списков L0, L1,..., L9 в общий список  
L = L0 => L1 => L2 => ... => L9

Рассмотрим пример работы алгоритма на входном списке  
0 => 8 => 12 => 56 => 7 => 26 => 44 => 97 => 2 => 37 => 4 => 3 => 3 => 45 => 10.

Максимальное число содержит две цифры, значит, разрядность данных k=2.

## Первый проход, j=1.     Распределение по первой справа цифре: L0: 0 => 10        // все числа с первой справа цифрой 0 L1: пусто L2: 12 => 2 L3: 3 => 3 L4: 44 => 4 L5: 45 L6: 56 => 26 L7: 7 => 97 => 37 L8: 8 L9: пусто         // все числа с первой справа цифрой 9     Cборка:  соединяем списки Li один за другим L: 0 => 10 => 12 => 2 => 3 => 3 => 44 => 4 => 45 => 56 => 26 => 7 => 97 => 37 => 8 Второй проход, j=2.     Распределение по второй справа цифре: L0: 0 => 2 => 3 => 3 => 4 => 7 => 8 L1: 10 => 12 L2: 26 L3: 37 L4: 44 => 45 L5: 56 L6: пусто L7: пусто L8: пусто L9: 97     Cборка:  соединяем списки Li один за другим L: 0 => 2 => 3 => 3 => 4 => 7 => 8 => 10 => 12 => 26 => 37 => 44 => 45 => 56 => 97

1. **Формальная постановка задачи.**
   1. *Исходные данные:*

* Элементы списка;
* Максимальная длина элемента;
  1. *Ограничения на исходные данные:*

-Вводимые элементы – натуральные числа.

* 1. *Результирующие данные:*
* Отсортированный список;
* Время выполнения функции

1. **Функция сортировки.**

// функция сортировки возвращает указатель на начало отсортированного списка

Node \*radix\_list(Node \*l, int t) {

// t - разрядность (максимальная длина числа)

int i, j, d, m=1;

Node \*temp, \*head[10], \*tail[10];

for ( j = 1; j <= t; j++ )

{

for ( i = 0; i <= 9; i++ )

head[i] = ( tail[i] = NULL );

while ( l != NULL )

{

d = ((int)(l->val/m))%10;

temp = tail[d];

if ( head[d] == NULL )

head[d] = l;

else

temp->next = l;

temp = tail[d] = l;

l = l->next;

temp->next = NULL;

}

for ( i = 0; i <= 9; i++ )

if ( head[i] != NULL ) break;

l = head[i];

temp = tail[i];

for (d=i+1; d<=9; d++) {

if ( head[d] != NULL) {

temp->next = head[d];

temp = tail[d];

}

}

m\*=10;

}

return (l);

}

1. **Набор тестов:**

|  |  |
| --- | --- |
| Вход | **Выход** |
| “ ” | Список пуст! |
| 2,45,67,1,0,45,123,1,999,0  M=3 | 0,0,1,1,2,45,45,67,123,999 |
| 3,23,6,56,4,1,76  M=1 | 1,3,23,4,56,6,76 |
| -4,3,32,-5,54,45  M=2 | 3,32,54,45 |

1. **Вывод.**

В ходе лабораторной работе была создана программа, которая сортирует списки методом поразрядной сортировки.