Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и

Информатики

СибГУТИ

Кафедра ПМиК

Лабораторная работа №13

«Цифровая сортировка»

Выполнил: студент 1 курса группы ИП-013

Иванов.Л.Д

Проверил: старший преподаватель кафедры ПМиК

Солодов П.С.

Новосибирск

2021

Лабораторная работа № 13

1). Разработать процедуру цифровой сортировки списка (DigitalSort). Для проверки правильности сортировки использовать сервисные процедуры для работы со списками, разработанные в лабораторной работе № 11.

2). Применить DigitalSort для сортировки двухбайтовых и четырехбайтовых целых чисел в прямом и обратном порядке.

**код:**

#include <cmath>

#include <ctime>

#include <iostream>

using namespace std;

int C = 0;

int M = 0;

struct Node {

Node \*next = nullptr;

union {

int data;

unsigned char digit[sizeof(int)];

};

};

struct Queue {

Node \*head;

Node \*tail;

};

void print\_list(Node \*head) {

Node \*p = head;

while (p != nullptr) {

cout << p->data << " ";

p = p->next;

}

cout << endl;

}

Node \*fill\_inc(int n) {

Node \*p = nullptr, \*head = nullptr;

for (int i = n - 1; i >= 0; i--) {

p = new Node;

p->data = i;

p->next = head;

head = p;

}

return head;

}

Node \*fill\_dec(int n) {

Node \*p = nullptr, \*head = nullptr;

for (int i = 0; i < n; i++) {

p = new Node;

p->data = i;

p->next = head;

head = p;

}

return head;

}

Node \*fill\_rand(int n) {

Node \*p = nullptr, \*head = nullptr;

srand(time(nullptr));

for (int i = 0; i < n; i++) {

p = new Node;

p->data = rand() % (n \* 2);

p->next = head;

head = p;

}

return head;

}

int check\_sum(Node \*head) {

int sum = 0;

Node \*p = head;

while (p) {

sum += p->data;

p = p->next;

}

return sum;

}

int run\_number(Node \*head) {

int i = 1;

Node \*p = head;

while (p->next) {

if (p->data > p->next->data) {

i += 1;

}

p = p->next;

}

return i;

}

void clear(Node \*&head) {

if (head == nullptr)

return;

Node \*p = head, \*temp;

while (p) {

temp = p;

p = p->next;

delete temp;

}

head = nullptr;

}

int get\_size(Node \*head) {

int size = 0;

Node \*p = head;

while (p) {

size++;

p = p->next;

}

return size;

}

void digital\_sort(Node \*&S) {

unsigned int L = sizeof(int);

Queue Q[256];

for (int j = L; j > -1; j--) {

for (int i = 0; i < 256; i++) {

Q[i].tail = (Node \*)&Q[i].head;

}

Node \*p = S;

while (p) {

unsigned char d = p->digit[j];

Q[(int)d].tail->next = p;

Q[(int)d].tail = p;

p = p->next;

}

p = (Node \*)&S;

for (int i = 0; i < 256; i++) {

if (Q[i].tail != (Node \*)&Q[i].head) {

p->next = Q[i].head;

p = Q[i].tail;

}

}

p->next = nullptr;

}

}

void digital\_sort\_reversed(Node \*&S) {

unsigned int L = sizeof(int);

Queue Q[256];

for (int j = L; j > -1; j--) {

for (int i = 0; i < 256; i++) {

Q[i].tail = (Node \*)&Q[i].head;

}

Node \*p = S;

while (p) {

unsigned char d = p->digit[j];

Q[(int)d].tail->next = p;

Q[(int)d].tail = p;

p = p->next;

}

p = (Node \*)&S;

for (int i = 256; i > -1; i--) {

if (Q[i].tail != (Node \*)&Q[i].head) {

p->next = Q[i].head;

p = Q[i].tail;

}

}

p->next = nullptr;

}

}

int main() {

Node \*list = fill\_rand(10);

digital\_sort(list);

print\_list(list);

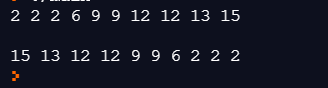
cout << "\n";

digital\_sort\_reversed(list);

print\_list(list);

return 0;

}



Вывод: В данной лабораторной работе, я изучил Цифровою сортировку и практически использовал пройденные темы в течении всего семестра.