Государственное образовательное учреждение высшего

профессионального образования

Московский технический университет связи и информатики

*КАФЕДРА ИНФОРМАТИКИ*

**ЛАБОРАТОРНАЯ РАБОТА №2**

Выполнил:

Зацепин Максим (БИБ2206)

Проверила: Степанова И.В.

2023г.

**Цели и задачи:** Изучить основные алгоритмы сортировки данных и освоить их на практике. Проверить и исследовать работу алгоритмов на различных наборах данных.

**Алгоритм сортировки - это алгоритм для упорядочивания элементов в**

**списке.**

**Задания:**

1. Разработать подпрограммы сортировки массива определенного

типа данных (unsigned int) с помощью алгоритмов сортировки:

- Сортировка выбором

- Сортировка расчёской

2. Отладить правильность работы сортировок на массивах с количеством элементов N=50 сгенерированные датчиком случайных чисел в диапазоне [Ix0x0, 3x70x], где хх - цифры номера варианта. Кроме того, контролировать правильность сортировки путем подсчета контрольной суммы и числа серий в массиве (оформить в виде подпрограммы).

*Серией называется неубывающая последовательность элементов массива максимальной длины. Пример: в массиве 23145314 (23 145 3 14) содержится 4 серии*

3. Для сравнения алгоритмов сортировки составить таблицу следующего вида (данные получить экспериментально) для N=100, 200, 300, 400, 500. (N - количество элементов в массиве). Для этого в программе обязательно вставляется точки для замера времени для выдачи времени выполнения алгоритма.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Количество  элементов | 50 | 100 | 200 | 300 | 400 | 500 |
| Алгоритм  сортировки | Время выполнения алгоритма сортировки (в миллисекундах) | | | | | |
| Выбором | 0.000124 | 0.000208 | 0.000207 | 0.000563 | 0.000457 | 0.000568 |
| Расчёской | 0.000146 | 0.000232 | 0.000316 | 0.000398 | 0.000298 | 0.000411 |

**Сортировка выбором:**

#include <iostream>

#include <time.h>

using namespace std;

// Массив из 50ти чисел [17070,37707]

// num = m + rand() % (n - m + 1);

void createArray(int n, unsigned int arr[]);

void sortByChoise(int n, unsigned int arr[]);

int getNextRatio(int ratio);

int main(){

srand(time(nullptr));

int n = 50;

unsigned int arr[50];

createArray(n,arr);

cout << "\n\nВывод неотсортированного массива\n";

for(int i = 0, count = 1; i < n; i++, count++){

cout << arr[i] << " ";

if(count == 10){ cout << endl; count = 0; }

}

cout << "\n\n";

sortByChoise(n,arr);

cout << "\n\nВывод отсортированного массива\n";

for(int i = 0, count = 1; i < n; i++, count++){

cout << arr[i] << " ";

if(count == 10){ cout << endl; count = 0; }

}

cout << "\n\n";

}

// Создание массива:

void createArray(int n, unsigned int arr[]){

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

arr[i] = 17070 + rand() % (37707 - 17070 + 1);

}

}

// Получить следующее расстояние между элементами

int getNextRatio(int ratio){

ratio = (ratio \* 10) / 13;

if(ratio < 1){ return 1; }

return ratio;

}

// Сортировка выбором

void sortByChoise(int n, unsigned int arr[]){

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

int minIndex = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

swap(arr[i], arr[minIndex]);

}

}

**Сортировка расчёской:**

#include <iostream>

#include <time.h>

using namespace std;

// Массив из 50ти чисел [17070,37707]

// num = m + rand() % (n - m + 1);

void createArray(int n, unsigned int arr[]);

void sortByBrush(int n, unsigned int arr[]);

int getNextRatio(int ratio);

int main(){

srand(time(nullptr));

int n = 50;

unsigned int arr[50];

createArray(n,arr);

cout << "\n\nВывод неотсортированного массива\n";

for(int i = 0, count = 1; i < n; i++, count++){

cout << arr[i] << " ";

if(count == 10){ cout << endl; count = 0; }

}

cout << "\n\n";

sortByBrush(n,arr);

cout << "\n\nВывод отсортированного массива\n";

for(int i = 0, count = 1; i < n; i++, count++){

cout << arr[i] << " ";

if(count == 10){ cout << endl; count = 0; }

}

cout << "\n\n";

}

// Создание массива:

void createArray(int n, unsigned int arr[]){

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

arr[i] = 17070 + rand() % (37707 - 17070 + 1);

}

}

// Получить следующее расстояние между элементами

int getNextRatio(int ratio){

ratio = (ratio \* 10) / 13;

if(ratio < 1){ return 1; }

return ratio;

}

// Сортировка расчёской

void sortByBrush(int n, unsigned int arr[]){

int sortRatio = n;

bool isSwapped = true;

while(sortRatio != 1 || isSwapped){

sortRatio = getNextRatio(sortRatio);

isSwapped = false;

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

if(arr[i] > arr[i + sortRatio]){

swap(arr[i],arr[i + sortRatio]);

isSwapped = true;

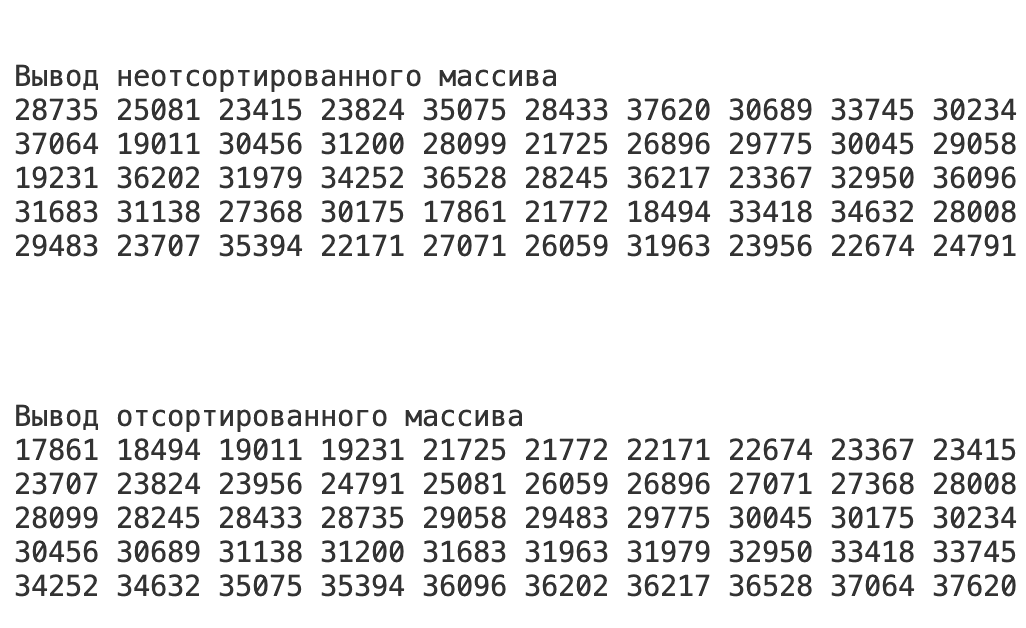
}

}

}

}

**Результат программы:**

****