**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

**отчет**

**по лабораторной работе №3**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: «Сортировка»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 0322 |  | Павлова А.Н. |
| Преподаватель |  | Пестерев Д.О. |

Санкт-Петербург

2022

1. Постановка задачи

Реализовать следующие алгоритмы сортировки:

1. Сортировка вставками (Insertion sort)
2. Сортировка выбором (Selection sort)
3. Пузырьковая сортировка (Bubble sort)
4. Сортировка слиянием (Merge sort)
5. Сортировка Шелла (Shell sort)
6. Быстрая сортировка (Quick sort)

Для каждого алгоритма сортировки указать временную асимптотическую сложность для лучшего, худшего случая и среднего случая, а также пространственную сложность, подкрепив это логическими построениями (*в меру своих сил*). Свести получившиеся результаты в таблицу. Построить график зависимости времени выполнения от размера входных данных и определить временную асимптотическую сложность для лучшего, худшего, среднего случая практически. Определить наиболее быстрый алгоритм сортировки. Сравнить скорость его выполнения с одним из алгоритмов сортировки из базовых библиотек языка.

1. Описание реализуемого класса и методов

Реализуемые методы:

* void insertionSort – сортировка вставками
* void choicesSort– сортировка выбором
* void bubbleSort– пузырьковая сортировка
* void ShellSort– сортировка Шелла
* void quickSort – быстрая сортировка

1. Оценка временной сложности каждого метода

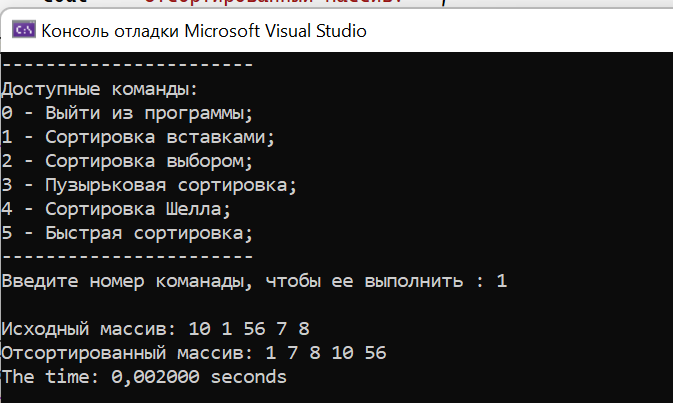
|  |  |
| --- | --- |
| метод | **Сложность** |
| void insertionSort | O(n^2) |
| void choicesSort | O(n^2) |
| void bubbleSort | O(n^2) |
| void ShellSort | O(n^2) |
| void quickSort | O(n log n) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Сортировка вставками | Сортировка выбором | Пузырьковая сортировка | Сортировка Шелла | Быстрая сортировка |
| Худший случай | O(n^2) | O(n^2) | O(n^2) | O(n^2) | O(n^2) |
| Лучший случай | O(n) | O(n^2) | O(n) | O(*n* log *n*) | O(*n* log *n*) |
| Средний случай | O(n^2) | O(n^2) | O(n^2) | Зависит от расстояния между элементами | O(*n* log *n*) |

Исходя из полученных данных, можем сделать вывод, что самый быстрой способ сортировки – сортировка Шелла.

1. Пример работы программы

Первый запуск.



При первом запуске программы мы выбираем какую сортировку хотим сделать, например, сортировка вставками. На выходе получаем данные исходный массив, отсортированный и время, которое было затрачено на сортировку.

Видим, что программа работает верно.

1. Листинг

**Заголовочные файлы:**

***Head.h***

#pragma once

#include <iostream>

#include <cstring>

#include <string>

#include <cmath>

#include <stdio.h>

#include <time.h>

void insertionSort(int arr[], int n);

void choicesSort(int arr[], int length\_array);

void bubbleSort(int arr[], int length\_array);

void ShellSort(int arr[], int length\_array);

void quickSort(int arr[], int length\_array);

***laba3.cpp***

#include "Head.h"

using namespace std;

void insertionSort(int arr[], int length\_array) { // сортировка вставками

int counter = 0;

for (int i = 1; i < length\_array; i++) {

for (int j = i; j > 0 && arr[j - 1] > arr[j]; j--) {

counter++;

int tmp = arr[j - 1];

arr[j - 1] = arr[j];

arr[j] = tmp;

}

}

cout << "Отсортированный массив: " ;

for (int i = 0; i < length\_array; i++) cout << arr[i] << " "; //вывод массива

}

void choicesSort(int arr[], int length\_array) // сортировка выбором

{

for (int repeat\_counter = 0; repeat\_counter < length\_array; repeat\_counter++)

{

int temp = arr[0]; // временная переменная для хранения значения перестановки

for (int element\_counter = repeat\_counter + 1; element\_counter < length\_array; element\_counter++)

{

if (arr[repeat\_counter] > arr[element\_counter])

{

temp = arr[repeat\_counter];

arr[repeat\_counter] = arr[element\_counter];

arr[element\_counter] = temp;

}

}

}

cout << "Отсортированный массив: " ;

for (int i = 0; i < length\_array; i++) cout << arr[i] << " "; //вывод массива

}

void bubbleSort(int arr[], int length\_array) // пузырьковая сортировка

{

for (int i = 0; i < length\_array; i++) {

for (int j = 0; j < (length\_array - 1); j++) {

if (arr[j] > arr[j + 1]) {

int b = arr[j]; // создали дополнительную переменную

arr[j] = arr[j + 1]; // меняем местами

arr[j + 1] = b; // значения элементов

}

}

}

cout << "Отсортированный массив: ";

for (int i = 0; i < length\_array; i++) cout << arr[i] << " "; //вывод массива

}

int i, j, d, schet;

void ShellSort(int arr[], int length\_array) //сортировка Шелла

{

d = length\_array;

d = d / 2;

while (d > 0)

{

for ( i = 0; i < length\_array - d; i++)

{

j = i;

while (j >= 0 && arr[j] > arr[j + d])

{

schet = arr[j];

arr[j] = arr[j + d];

arr[j + d] = schet;

j--;

}

}

d = d / 2;

}

cout << "Отсортированный массив: ";

for ( int i = 0; i < length\_array; i++) cout << arr[i] << " "; //вывод массива

}

void quickSort(int arr[], int length\_array) { // быстрая сортировка

//Указатели в начало и в конец массива

int i = 0;

int j = length\_array - 1;

//Центральный элемент массива

int mid = arr[length\_array / 2];

//Делим массив

do {

//Пробегаем элементы, ищем те, которые нужно перекинуть в другую часть

//В левой части массива пропускаем(оставляем на месте) элементы, которые меньше центрального

while (arr[i] < mid) {

i++;

}

//В правой части пропускаем элементы, которые больше центрального

while (arr[j] > mid) {

j--;

}

//Меняем элементы местами

if (i <= j) {

int tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

i++;

j--;

}

} while (i <= j);

//Рекурсивные вызовы, если осталось, что сортировать

if (j > 0) {

//"Левый кусок"

quickSort(arr, j + 1);

}

if (i < length\_array) {

//"Прaвый кусок"

quickSort(&arr[i], length\_array - i);

}

cout << "Отсортированный массив: ";

for (int i = 0; i < length\_array; i++) cout << arr[i] << " "; //вывод массива

}

int main() {

setlocale(LC\_ALL, "Russian");

const int n=5;

int Arr[n] = {10,1,56,7,8};

int Numb = 0;

cout << "-----------------------" << endl;

cout << "Доступные команды: " << endl;

cout << "0 - Выйти из программы;" << endl;

cout << "1 - Сортировка вставками;" << endl;

cout << "2 - Сортировка выбором;" << endl;

cout << "3 - Пузырьковая сортировка;" << endl;

cout << "4 - Сортировка Шелла;" << endl;

cout << "5 - Быстрая сортировка;" << endl;

cout << "-----------------------" << endl;

cout << "Введите номер команады, чтобы ее выполнить : ";

cin >> Numb;

while (Numb != 0)

{

switch (Numb)

{

case 0: {

cout << "Спасибо, что воспользовались нашей программой" << endl;

break;

}

case 1: { //сортировка вставками

cout <<endl<< "Исходный массив: ";

for (i = 0; i < 5; i++) {

cout << Arr[i] << " ";

}

cout << endl;

clock\_t start = clock();

insertionSort(&Arr[0], n);

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

cout << endl;

printf("The time: %f seconds\n", seconds);

Numb = 0;

break;

}

case 2: {// сортировка выбором

cout <<endl<< "Исходный массив: " ;

for (i = 0; i < 5; i++) {

cout << Arr[i] << " ";

}

cout << endl;

clock\_t start = clock();

choicesSort(&Arr[0], n);

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

cout << endl;

printf("The time: %f seconds\n", seconds);

Numb = 0;

break;

}

case 3: {// пузырьковая сортировка

cout <<endl<< "Исходный массив: " ;

for (i = 0; i < 5; i++) {

cout << Arr[i] << " ";

}

cout << endl;

clock\_t start = clock();

bubbleSort(&Arr[0], n);

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

cout << endl;

printf("The time: %f seconds\n", seconds);

Numb = 0;

break;

}

case 4: {// сортировка Шелла

cout <<endl<< "Исходный массив: ";

for (i = 0; i < 5; i++) {

cout << Arr[i] << " ";

}

cout << endl;

clock\_t start = clock();

ShellSort(&Arr[0], n);

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

cout << endl;

printf("The time: %f seconds\n", seconds);

Numb = 0;

break;

}

case 5: {// быстрая сортировка

cout <<endl << "Исходный массив: " ;

for (i = 0; i < 5; i++) {

cout << Arr[i] << " ";

}

cout << endl;

clock\_t start = clock();

quickSort(&Arr[0], n);

clock\_t end = clock();

double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;

cout << endl;

printf("The time: %f seconds\n", seconds);

Numb = 0;

break;

}

}

}

}

1. Ссылка на репозиторий

https://github.com/typicalanya/lab3sem5