|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное учреждение  высшего образования | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Кафедра прикладной математики, механики, управления и программного обеспечения** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | |
| по лабораторной работе №1.3  дисциплина «Фундаментальные структуры данных и алгоритмы» | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Студент гр. Б9120-09.03.04прогин | | | |
|  |  |  | | О. Г. Александрович | |
|  | | | | | | |  |  | (подпись) | |  | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Руководитель | | |  |
|  | | | | | | |  |  | ст. преподаватель | | | |
|  | | | |  |  | |  |  |  |  | О.А. Крестникова | |
|  | | | |  |  | |  |  | (подпись) |  | (И.О. Фамилия) | |
|  |  |  |  | | |  |  |  |  | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | |
| 2022 | | | | | | | | | | | | |

Оглавление

[Неформальная постановка задачи: 2](#_Toc92386898)

[Постановка задачи: 2](#_Toc92386899)

[Описание алгоритма сортировки 3](#_Toc92386900)

[Сортировка Двухпутевыми вставками 3](#_Toc92386901)

[Быстрая сортировка 3](#_Toc92386902)

[Текст программы 3](#_Toc92386903)

[Пример входного файла (на 200 записей): 20](#_Toc92386904)

[Пример выходного файла (на 200 записей): 24](#_Toc92386905)

[Тесты 28](#_Toc92386906)

[1. 20 случайно сгенерированных записей: 28](#_Toc92386907)

[2. 200 случайно сгенерированных записей: 28](#_Toc92386908)

[3. 2000 случайно сгенерированных записей: 28](#_Toc92386909)

[Результат: 29](#_Toc92386910)

[Примеры, на которых достигается худший и лучший случаи 29](#_Toc92386911)

[1. Входные данные уже отсортированы: 29](#_Toc92386912)

[2. Входные данные отсортированы в обратном порядке: 29](#_Toc92386913)

[3. Входные данные содержат 100 одинаковых записей: 29](#_Toc92386914)

[Результат: 29](#_Toc92386915)

[Проверка устойчивости с обоснованием результатов 29](#_Toc92386916)

[Результат: 30](#_Toc92386917)

[Выводы: 30](#_Toc92386918)

# Неформальная постановка задачи:

Сортировать по убыванию, алгоритмы: Двухпутевые вставки, быстрая (не рекурсивная реализация (с использованием статического стека), опорный – средний).

В строке файла: ФИО, должность, дата приема на работу, стаж

Сортировать по полям: должность, стаж, ФИО

# Постановка задачи:

Входные данные: текстовый файл, в первой строке – целое число n - количество строк файла (10 ≤ n ≤ 1000000). Каждая из последующих строк содержит данные заданной предметной области (см. свой вариант задания), разделенные 1 пробелом. Все данные во входном файле корректные.

Выходные данные: текстовый файл, в первой строке - время, затраченное на сортировку входного файла, каждая из последующих строк содержит отсортированные данные в табличном виде (см. свой вариант задания).

1. Отсортировать данные входного файла в соответствии с заданным порядком сортировки (см. свой вариант задания)

2. Проверить на устойчивость (привести примеры, доказывающие, что сортировка неустойчивая)

3. Сравнить по времени (без учета чтения из файла и записи в файл)

4. Определить последовательности, на которых достигается наихудшее и наилучшее значение времени

# Описание алгоритма сортировки

## Сортировка Двухпутевыми вставками

Этот метод является модификацией сортировки простыми вставками. Резервируется некоторый выходной массив размером п, и первый элемент исходной последовательности помещается в середину резервного массива. Когда в середине резервного массива находится непрерывная группа элементов, место для нового элемента отводится при помощи сдвига всех элементов с меньшими значениями ключей на один шаг влево или элементов с большими значениями ключей на один шаг вправо. Выбор, в какую сторону выполнять сдвиг, зависит от того, какой из них приведет к меньшему количеству сдвигов.

## Быстрая сортировка

Нерекурсивная реализация быстрой сортировки использует явный стек магазинного типа, заменяя рекурсивные вызовы помещением в стек параметров, а рекурсивные вызовы процедур и выходы из них - циклом, выталкивающим параметры из стека и обрабатывающим их, пока стек не пуст. Больший из двух подфайлов помещается в стек первым, чтобы максимальная глубина стека при сортировке N элементов не превосходила lgN

# Текст программы

// Я не разделил этот файл на заголовочный и несколько основных, потому что это вызывает неустранимую ошибку:

// "Ссылка на неразрешённый внешний символ" - и указывает на первую строку кода.

// Поиск в интернете по названию и коду ошибки результатов не дал.

#include <iostream>

#include <cmath>

#include <time.h>

#include <fstream>

#include <stdlib.h>

#include <iomanip>

#include <string>

#include <Windows.h>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

#include <errno.h>

using namespace std;

#define MAXSTACK 2048 // максимальный размер стека

#pragma warning(disable : 4996)

int const AllCount = 2000; // Столько записей мы будем создавать и сортировать

bool isCreateNewDataFile = true; // Если true, то создастся новый файл данных, и уже он будет отсотрирован

// Если false - то будет использоватся имеющийся файл данных. Если его не найдётся - выведется ошибка.

bool mainCreateData(int AllCount) // Генерируем данные текстового файла из 4х других текстовых файлов

{

const int size = 250; // Максимальная длинна строки (Имени или профессии)

string S[size];

string str;

int n = 0;

int r;

bool genderMan = true;

fstream OutFile("OutFile.txt", ios::out); // открываем файл для записи в конец //app - в конец файла

cout << "count = " << AllCount << endl;

int turCount = 0;

while (turCount < AllCount)

{

turCount++;

fstream Famil("Rus Famil.txt", fstream::in);

fstream ManName("Rus MenName.txt", fstream::in);

fstream WomenName("Rus FaName.txt", fstream::in);

fstream Prof("Proffesion.txt", fstream::in);

if (Famil.is\_open() && ManName.is\_open() && WomenName.is\_open() && Prof.is\_open())

{

if ((rand() % 2 + 0) == 1)

{

genderMan = false;

}

r = rand() % 20 + 0;

n = 0;

while (!Famil.eof())

{

getline(Famil, S[n]);

if (n == r)

{

if (genderMan == true) OutFile << S[n] << " ";

else OutFile << S[n] << "а" << " ";

break;

}

n++;

}

if (genderMan == true)

{

r = rand() % 25 + 0;

n = 0;

while (!ManName.eof())

{

getline(ManName, S[n]);

if (n == r)

{

OutFile << S[n] << " ";

break;

}

n++;

}

ManName.close(); WomenName.close();

fstream ManName("Rus MenName.txt", fstream::in);

fstream WomenName("Rus FaName.txt", fstream::in);

r = rand() % 25 + 0;

n = 0;

while (!ManName.eof())

{

getline(ManName, S[n]);

if (n == r)

{

OutFile << S[n] << "ович";

break;

}

n++;

}

}

else

{

r = rand() % 25 + 0;

n = 0;

while (!WomenName.eof())

{

getline(WomenName, S[n]);

if (n == r)

{

OutFile << S[n] << " ";

break;

}

n++;

}

ManName.close(); WomenName.close();

fstream ManName("Rus MenName.txt", fstream::in);

fstream WomenName("Rus FaName.txt", fstream::in);

r = rand() % 25 + 0;

n = 0;

while (!WomenName.eof())

{

getline(WomenName, S[n]);

if (n == r)

{

OutFile << S[n] << "овна";

break;

}

n++;

}

}

OutFile << ", ";

r = rand() % 25 + 0;

n = 0;

while (!Prof.eof())

{

getline(Prof, S[n]);

if (n == r)

{

OutFile << S[n];

break;

}

n++;

}

//22.08.2020

int dd, mm, gg;

dd = rand() % 29 + 1;

mm = 11; //rand() % 11 + 1;

gg = 2021; //rand() % 20 + 2000;

OutFile << ", " << dd << "." << mm << "." << gg;

int MainDat = 2022 \* 12 \* 30;

int AllDat = gg \* 12 \* 30 + mm \* 30 + dd;

int currStaj = MainDat - AllDat;

OutFile << ", " << currStaj << " дней";

OutFile << endl;

Famil.close(); ManName.close(); WomenName.close(); Prof.close();

}

else return false;

}

OutFile.close();

return true;

}

struct Data

{

int dd;

int mm;

int gg;

};

Data data;

struct Dann

{

string Name;

string Famil;

string Otth;

string Proff;

Data Data;

int Staj;

};

Dann dann;

bool Processing\_data\_from\_the\_input\_file(Dann\* mass)

// Распознаём и записываем данные из файла во внутреннй массив программы, для дальнеёшей сортировки

{

string S[AllCount];

int n = 0;

if (isCreateNewDataFile == true)

{

if (mainCreateData(AllCount)) // Генерируем записи

{

cout << "Данные успешно сгенерированы!" << endl;

cout << endl;

}

else

{

return false;

}

}

fstream DataFile("OutFile.txt", fstream::in);

if (DataFile.is\_open())

{

string str = "";

int intMass = 0;

n = 0;

while (n < AllCount) //(!DataFile.eof())

{

getline(DataFile, S[n]);

cout << S[n] << endl;

string input = S[n];

char r;

int y = 0;

str = "";

r = input[y];

while (r != ' ')

{

r = input[y];

if (r != ' ')

{

str = str + r;

}

y++;

}

//cout << "Имя (str) = " << str << endl;

mass[intMass].Name = str;

str = "";

r = input[y];

while (r != ' ')

{

r = input[y];

if (r != ' ')

{

str = str + r;

}

y++;

}

//cout << "Фамилия (str) = " << str << endl;

mass[intMass].Famil = str;

str = "";

r = input[y];

while (r != ',')

{

r = input[y];

if (r != ',')

{

str = str + r;

}

y++;

}

//cout << "Отество (str) = " << str << endl;

mass[intMass].Otth = str;

str = "";

r = input[y];

while (r != ',')

{

r = input[y];

if (r != ',')

{

str = str + r;

}

y++;

}

//cout << "Профессия (str) = " << str << endl;

mass[intMass].Proff = str;

str = "";

r = input[y];

while (r != '.')

{

r = input[y];

if (r != '.')

{

str = str + r;

}

y++;

}

//cout << "Число (str) = " << str << endl;

char\* strng = new char[str.length() + 1];

strcpy(strng, str.c\_str());

mass[intMass].Data.dd = atoi(strng);

str = "";

r = input[y];

while (r != '.')

{

r = input[y];

if (r != '.')

{

str = str + r;

}

y++;

}

//cout << "Месяц (str) = " << str << endl;

strng = new char[str.length() + 1];

strcpy(strng, str.c\_str());

mass[intMass].Data.mm = atoi(strng);

str = "";

r = input[y];

while (r != ',')

{

r = input[y];

if (r != ',')

{

str = str + r;

}

y++;

}

//cout << "Год (str) = " << str << endl;

strng = new char[str.length() + 1];

strcpy(strng, str.c\_str());

mass[intMass].Data.gg = atoi(strng);

str = "";

y++;

r = input[y];

while (r != ' ')

{

r = input[y];

if (r != ' ')

{

str = str + r;

}

y++;

}

//cout << "Стаж (str) = " << str << endl;

strng = new char[str.length() + 1];

strcpy(strng, str.c\_str());

mass[intMass].Staj = atoi(strng);

n++;

intMass++;

}

}

else return false;

DataFile.close();

return true;

}

Dann\* DataCreator()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Dann\* mass = new Dann[AllCount]; // Массив для храниения записей как структуру полей

if (Processing\_data\_from\_the\_input\_file(mass)) // Загружаем во внутренний массив записи из файла

{

cout << endl;

cout << "Данные успешно обработаны!" << endl;

return mass;

//outputDann(mass[19]);

}

else

{

cout << "Файл, необходимый для обработки записей, не открывается." << endl;

return NULL;

}

}

// Далее я создаю другую структуру массива данных, и перезаписываю в неё все данные из моего 1го массива.

// Это делается потому, что код генерации и распознавания данных, и код сортировки данных я писал отдельно друг от друга

struct AQ // Стандартная структура нужных нам данных

{

string d; // Должность

int s; // Стаж

string f; // Фамилия

string i; // Имя

string o; // Отчество

};

struct AQ2 // Структура данны, с дополнительным индексом, для проверки устойчивости сортировок

{

string d; // Должность

int s; // Стаж

string f; // Фамилия

string i; // Имя

string o; // Отчество

int ind;

};

// У нас сортировка по: Должность, Стаж, Фамилия, Имя, Отчество (5шт.)

bool RepeatAZ\_1(AQ a, AQ b) // if(a >= b)

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return true;

}

else if (a.o > b.o) return true;

else return false;

}

else if (a.i > b.i) return true;

else return false;

}

else if (a.f > b.f) return true;

else return false;

}

else if (a.s > b.s) return true;

else return false;

}

else if (a.d > b.d) return true;

else return false;

}

bool RepeatAZ\_2(AQ a, AQ b) // if(a < b)

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return false;

}

else if (a.o < b.o) return true;

else return false;

}

else if (a.i < b.i) return true;

else return false;

}

else if (a.f < b.f) return true;

else return false;

}

else if (a.s < b.s) return true;

else return false;

}

else if (a.d < b.d) return true;

else return false;

}

bool RepeatAZ\_3(AQ a, AQ b) // if(a > b)

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return false;

}

else if (a.o > b.o) return true;

else return false;

}

else if (a.i > b.i) return true;

else return false;

}

else if (a.f > b.f) return true;

else return false;

}

else if (a.s > b.s) return true;

else return false;

}

else if (a.d > b.d) return true;

else return false;

}

void Tinsert(AQ\* InputMass, int n) // Сортировка Двухпутевыми вставками

{ // n - размер исходного массива

int i, j, left, right, newN;

AQ t;

left = n - 1; // Устанавливаем указатели на правую и левую границу в середину нового массива

right = n - 1; // И сдвигаем их по мере добавления элементов

newN = 2 \* n;

AQ\* NewMass = new AQ[newN];

NewMass[n - 1] = InputMass[0]; // Центральный элемент расширенного массива - начальный исходного массива.

for (i = 0; i < n; i++) // Проходим исходный массив слева направо

{

t = InputMass[i];

if (RepeatAZ\_1(t, InputMass[0])) // if (t >= InputMass[0])

{

for (j = right; (j >= 0) && RepeatAZ\_2(t, NewMass[j]); j--)

NewMass[j + 1] = NewMass[j];

// Сдвигаем все значения массива вправо (от правой границы), пока наше взятое значение не станет больше итерационного

NewMass[j + 1] = t;

right++;

}

else

{

for (j = left; (j <= newN) && RepeatAZ\_3(t, NewMass[j]); j++)

NewMass[j - 1] = NewMass[j];

// Сдвигаем все значения массива влево (от левой границы), пока наше взятое значение не станет меньше итерационного

NewMass[j - 1] = t;

left--;

}

}

for (j = left+1; j < left+1 + n; j++) // Перезаписываем отсортированные значения в начальный массив

InputMass[j - left -1] = NewMass[j];

delete[] NewMass;

}

bool RepeatAZ\_1(AQ2 a, AQ2 b) // if(a >= b) // Для примера неустойчивой сотрировки

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return true;

}

else if (a.o > b.o) return true;

else return false;

}

else if (a.i > b.i) return true;

else return false;

}

else if (a.f > b.f) return true;

else return false;

}

else if (a.s > b.s) return true;

else return false;

}

else if (a.d > b.d) return true;

else return false;

}

bool RepeatAZ\_2(AQ2 a, AQ2 b) // if(a < b) // Для примера неустойчивой сотрировки

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return false;

}

else if (a.o < b.o) return true;

else return false;

}

else if (a.i < b.i) return true;

else return false;

}

else if (a.f < b.f) return true;

else return false;

}

else if (a.s < b.s) return true;

else return false;

}

else if (a.d < b.d) return true;

else return false;

}

bool RepeatAZ\_3(AQ2 a, AQ2 b) // if(a > b) // Для примера неустойчивой сотрировки

{

if (a.d == b.d)

{

if (a.s == b.s)

{

if (a.f == b.f)

{

if (a.i == b.i)

{

if (a.o == b.o)

{

return false;

}

else if (a.o > b.o) return true;

else return false;

}

else if (a.i > b.i) return true;

else return false;

}

else if (a.f > b.f) return true;

else return false;

}

else if (a.s > b.s) return true;

else return false;

}

else if (a.d > b.d) return true;

else return false;

}

void Tinsert2(AQ2\* InputMass, int n) // Сортировка Двухпутевыми вставками

{ // n - размер исходного массива

int i, j, left, right, newN;

AQ2 t;

left = n - 1; // Устанавливаем указатели на правую и левую границу в середину нового массива

right = n - 1; // И сдвигаем их по мере добавления элементов

newN = 2 \* n;

AQ2\* NewMass = new AQ2[newN];

NewMass[n - 1] = InputMass[0]; // Центральный элемент расширенного массива - начальный исходного массива.

for (i = 0; i < n; i++) // Проходим исходный массив слева направо

{

t = InputMass[i];

if (RepeatAZ\_1(t, InputMass[0])) // if (t >= InputMass[0])

{

for (j = right; (j >= 0) && RepeatAZ\_2(t, NewMass[j]); j--)

NewMass[j + 1] = NewMass[j];

// Сдвигаем все значения массива вправо (от правой границы), пока наше взятое значение не станет больше итерационного

NewMass[j + 1] = t;

right++;

}

else

{

for (j = left; (j <= newN) && RepeatAZ\_3(t, NewMass[j]); j++)

NewMass[j - 1] = NewMass[j];

// Сдвигаем все значения массива влево (от левой границы), пока наше взятое значение не станет меньше итерационного

NewMass[j - 1] = t;

left--;

}

}

for (j = left; j < left + n; j++) // Перезаписываем отсортированные значения в начальный массив

InputMass[j - left] = NewMass[j];

delete[] NewMass;

}

void coutM(AQ\* mass, int n) // Выводим данные в консоль

{

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

{

cout << "[" << mass[i].d << ", " << mass[i].s << ", " << mass[i].f << ", " << mass[i].i << ", " << mass[i].o << "]" << endl;

}

}

void qSortI(AQ a[], long size) // Опорный - средний

{

long i, j; // указатели, участвующие в разделении

long lb, ub; // границы сортируемого в цикле фрагмента

long lbstack[MAXSTACK], ubstack[MAXSTACK]; // стек запросов

// каждый запрос задается парой значений,

// а именно: левой(lbstack) и правой(ubstack)

// границами промежутка

long stackpos = 1; // текущая позиция стека

long ppos; // середина массива

AQ pivot; // опорный элемент

// int temp;

lbstack[1] = 0;

ubstack[1] = size - 1;

do {

// Взять границы lb и ub текущего массива из стека.

lb = lbstack[stackpos];

ub = ubstack[stackpos];

stackpos--;

do {

// Шаг 1. Разделение по элементу pivot

ppos = (lb + ub) >> 1;

i = lb; j = ub; pivot = a[ppos];

do {

while (RepeatAZ\_2(a[i], pivot)) i++; // (a[i] < pivot)

while (RepeatAZ\_2(pivot, a[j])) j--; // (pivot < a[j])

if (i <= j)

{

//swapp(a[i], a[j]);

AQ temp; temp = a[i]; a[i] = a[j]; a[j] = temp;

i++; j--;

}

} while (i <= j);

// Шаги 2, 3. Отправляем большую часть в стек и двигаем lb,ub

if (i < ppos) { // правая часть больше

if (i < ub) { // если в ней больше 1 элемента - нужно

stackpos++; // сортировать, запрос в стек

lbstack[stackpos] = i;

ubstack[stackpos] = ub;

}

ub = j; // следующая итерация разделения

// будет работать с левой частью

}

else { // левая часть больше

if (j > lb) {

stackpos++;

lbstack[stackpos] = lb;

ubstack[stackpos] = j;

}

lb = i;

}

} while (lb < ub); // пока в меньшей части более 1 элемента

} while (stackpos != 0); // пока есть запросы в стеке

}

void qSortI(AQ2 a[], long size) // Опорный - средний // Для примера неустойчивой сотрировки

{

long i, j; // указатели, участвующие в разделении

long lb, ub; // границы сортируемого в цикле фрагмента

long lbstack[MAXSTACK], ubstack[MAXSTACK]; // стек запросов

long stackpos = 1; // текущая позиция стека

long ppos; // середина массива

AQ2 pivot; // опорный элемент

// int temp;

lbstack[1] = 0;

ubstack[1] = size - 1;

do {

// Взять границы lb и ub текущего массива из стека.

lb = lbstack[stackpos];

ub = ubstack[stackpos];

stackpos--;

do {

// Шаг 1. Разделение по элементу pivot

ppos = (lb + ub) >> 1;

i = lb; j = ub; pivot = a[ppos];

do {

while (RepeatAZ\_2(a[i], pivot)) i++; // (a[i] < pivot)

while (RepeatAZ\_2(pivot, a[j])) j--; // (pivot < a[j])

if (i <= j)

{

//swapp(a[i], a[j]);

AQ2 temp; temp = a[i]; a[i] = a[j]; a[j] = temp;

i++; j--;

}

} while (i <= j);

// Шаги 2, 3. Отправляем большую часть в стек и двигаем lb,ub

if (i < ppos) { // правая часть больше

if (i < ub) { // если в ней больше 1 элемента - нужно

stackpos++; // сортировать, запрос в стек

lbstack[stackpos] = i;

ubstack[stackpos] = ub;

}

ub = j; // следующая итерация разделения

// будет работать с левой частью

}

else { // левая часть больше

if (j > lb) {

stackpos++;

lbstack[stackpos] = lb;

ubstack[stackpos] = j;

}

lb = i;

}

} while (lb < ub); // пока в меньшей части более 1 элемента

} while (stackpos != 0); // пока есть запросы в стеке

}

void Examples\_Instability() // Процедура, которая записывает в файл примеры неустойчивости

{

fstream OutSortFile("\_OutSortFile.txt", ios::app);

AQ2\* massEI = new AQ2[2];

OutSortFile << "Введём новую структуру записи, содержащей поле 'индекс'" << endl;

OutSortFile << "Заполним массив из 2х записей одинаковыми значениями, с разными индексами" << endl;

OutSortFile << "Отсортируем 2мя разными способами и проверим, устойчива ли каждая сортировка или нет:" << endl;

for (int i = 0; i < 2; i++) // Заполним массив из 2х записей одинаковыми значениями:

{

// Крестникова Ольга Александровна

massEI[i].d = "Учитель";

massEI[i].i = "Ольга";

massEI[i].s = 10;

massEI[i].f = "Крестникова";

massEI[i].o = "Александровна";

}

massEI[0].ind = 0;

massEI[1].ind = 1;

AQ2\* massDP = new AQ2[2]; // Двухпутевыми

AQ2\* massQ = new AQ2[2]; // Быстрой

for (int i = 0; i < 2; i++)

{

massQ[i] = massEI[i];

massDP[i] = massEI[i];

}

OutSortFile << "Выводим данные без изменений: " << endl;

OutSortFile << endl;

for (int i = 0; i < 2; i++)

{

OutSortFile << "[" << massEI[i].d << ", " << massEI[i].s << ", " << massEI[i].f << ", " << massEI[i].i << ", " << massEI[i].o << "], ind = " << massEI[i].ind << endl;

}

OutSortFile << endl;

OutSortFile << "Сортируем Двухпутевыми вставками" << endl; // Устойчивая

Tinsert2(massDP, 2);

OutSortFile << "Выводим данные отсортированные Двухпутевыми вставками: " << endl;

OutSortFile << endl;

{

OutSortFile << "[" << massDP[0].d << ", " << massDP[0].s << ", " << massDP[0].f << ", " << massDP[0].i << ", " << massDP[0].o << "], ind = 0" << endl;

OutSortFile << "[" << massDP[1].d << ", " << massDP[1].s << ", " << massDP[1].f << ", " << massDP[1].i << ", " << massDP[1].o << "], ind = 1" << endl;

}

OutSortFile << endl;

OutSortFile << "Сортируем Быстрой сортировкой" << endl; // Не устойчивая

qSortI(massQ, 2);

OutSortFile << "Выводим данные отсортированные Быстрой сортировкой: " << endl;

OutSortFile << endl;

for (int i = 0; i < 2; i++)

{

OutSortFile << "[" << massQ[i].d << ", " << massQ[i].s << ", " << massQ[i].f << ", " << massQ[i].i << ", " << massQ[i].o << "], ind = " << massQ[i].ind << endl;

}

OutSortFile << endl;

OutSortFile << endl;

(OutSortFile).close();

delete[] massDP;

delete[] massQ;

delete[] massEI;

}

bool PrintToFile(AQ\* mass, int n, int SortTime, bool q) // Процедура, которая записыват выходные данные в файл. Вызывается 2 раза

{

fstream OutSortFile("\_OutSortFile.txt", ios::app);

if (q == false)

{

OutSortFile << "Количество записей: " << n << endl;

OutSortFile << endl;

OutSortFile << "Сортировка Двухпутевыми вставками: " << endl;

OutSortFile << "Время, затраченное на сортировку: " << SortTime << " милисекунд" << endl;

OutSortFile << endl;

return(false);

}

else

{

OutSortFile << "Сортировка Быстрой сортировкой: " << endl;

OutSortFile << "Время, затраченное на сортировку: " << SortTime << " милисекунд" << endl;

OutSortFile << endl;

}

OutSortFile << "Результат сортировки (отсортированные данные): " << endl;

OutSortFile << endl;

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

{

OutSortFile << "[" << mass[i].d << ", " << mass[i].s << ", " << mass[i].f << ", " << mass[i].i << ", " << mass[i].o << "]" << endl;

}

OutSortFile << endl;

OutSortFile << endl;

Examples\_Instability();

(OutSortFile).close();

return(true);

}

void main()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

AQ\* massDP = new AQ[AllCount]; // Двухпутевыми

AQ\* massQ = new AQ[AllCount]; // Быстрой

AQ\* massNS = new AQ[AllCount]; // Не отсортированный

Dann\* massA = new Dann[AllCount];

massA = DataCreator();

int SortTime = 0;

if (massA == NULL)

{

cout << "Данные были получены нерпавильно. Программа завершается с ошибкой." << endl;

}

else

{

// string d; // Должность

// int s; // Стаж

// string f; // Фамилия

// string i; // Имя

// string o; // Отчество

for (int i = 0; i < AllCount; i++) // Заполняем массив значениями

{

massNS[i].d = massA[i].Proff;

massNS[i].o = massA[i].Otth;

massNS[i].s = massA[i].Staj;

massNS[i].f = massA[i].Famil;

massNS[i].i = massA[i].Name;

}

fstream OutSortFile("\_OutSortFile.txt", ios::out); // Очищаем выходной файл

(OutSortFile).close();

cout << endl;

cout << "Выводим загруженные данные:" << endl;

for (int i = 0; i < AllCount; i++) // Копируем данные из неотсортированного массива в 2 других, для сортировки

{

massQ[i] = massNS[i];

massDP[i] = massNS[i];

}

coutM(massNS, AllCount);

// Измерение времени сортировки Двухпутевыми вставками

{

\_\_int64 start, end, tps;

QueryPerformanceFrequency((LARGE\_INTEGER\*)&tps);

QueryPerformanceCounter((LARGE\_INTEGER\*)&start);

Tinsert(massDP, AllCount); // Сортируем Двухпутевыми вставками

QueryPerformanceCounter((LARGE\_INTEGER\*)&end);

SortTime = ((double)(end - start) / tps) \* 1000.;

}

cout << endl;

cout << "Выводим данные, отсортированные Двухпутевыми вставками: " << endl;

coutM(massDP, AllCount);

PrintToFile(massDP, AllCount, SortTime, false);

cout << endl;

cout << "Время сортировки Двухпутевыми вставками: " << SortTime << " милисекунд";

cout << endl;

// ---------

// Измерение времени Быстрой сортировки

{

\_\_int64 start, end, tps;

QueryPerformanceFrequency((LARGE\_INTEGER\*)&tps);

QueryPerformanceCounter((LARGE\_INTEGER\*)&start);

qSortI(massQ, AllCount); // Сортируем Быстрой сортировкой

QueryPerformanceCounter((LARGE\_INTEGER\*)&end);

SortTime = ((double)(end - start) / tps) \* 1000.;

}

cout << endl;

cout << "Выводим данные, отсортированные Быстрой сортировкой: " << endl;

coutM(massQ, AllCount);

PrintToFile(massQ, AllCount, SortTime, true); // Печатаем данные в выходной файл

cout << endl;

cout << "Время сортировки Быстрой сортировкой: " << SortTime << " милисекунд";

cout << endl;

delete[] massDP;

delete[] massQ;

delete[] massNS;

delete[] massA;

}

}

# Пример входного файла (на 200 записей):

Файл генерируется автоматически, в программе, используя 4 других текстовых файла. Если не нужно генерировать новый файл с данными, а нужно использовать уже существующий, следует заменить значение переменной isCreateNewDataFile на false.

Воробьёва Полина Полинаовна, Археолог, 26.11.2021, 4 дней

Соколова Ирина Людмилаовна, Повар, 22.11.2021, 8 дней

Виноградова Александра Елизаветаовна, Актёр, 5.11.2021, 25 дней

Лебедева София Людмилаовна, Дизайнер ландшафта, 6.11.2021, 24 дней

Попова Ангелина Натальяовна, Микробиолог, 20.11.2021, 10 дней

Иванова Кристина Натальяовна, IT - менеджер, 5.11.2021, 25 дней

Фёдорова Полина Натальяовна, Инженер, 18.11.2021, 12 дней

Попова Ольга Юлияовна, Банкир, 10.11.2021, 20 дней

Петрова Надежда Екатеринаовна, Повар, 13.11.2021, 17 дней

Лебедева Ангелина Лидияовна, Пожарный, 29.11.2021, 1 дней

Васильева Лидия Натальяовна, IT - менеджер, 6.11.2021, 24 дней

Волкова Кира Алисаовна, Композитор, 22.11.2021, 8 дней

Петрова Нина Еленаовна, Банкир, 6.11.2021, 24 дней

Воробьёва Ангелина Александраовна, Звукооператор, 17.11.2021, 13 дней

Соколова Александра Вераовна, Руководитель, 18.11.2021, 12 дней

Козлова Елизавета Василисаовна, Авиадиспетчер, 2.11.2021, 28 дней

Иванова Надежда Викторияовна, Продюсер, 8.11.2021, 22 дней

Фёдорова София Елизаветаовна, Хирург, 27.11.2021, 3 дней

Виноградова Елена Екатеринаовна, Банкир, 28.11.2021, 2 дней

Попова Людмила Надеждаовна, Врач, 8.11.2021, 22 дней

Иванова Елена Анастасияовна, Авиадиспетчер, 6.11.2021, 24 дней

Воробьёва Ольга Викторияовна, IT - менеджер, 15.11.2021, 15 дней

Попова Наталья Юлияовна, Генетик, 6.11.2021, 24 дней

Козлова Людмила Вераовна, Хирург, 22.11.2021, 8 дней

Лебедева Светлана Светланаовна, Продюсер, 9.11.2021, 21 дней

Попова Вера Полинаовна, IT - менеджер, 26.11.2021, 4 дней

Кузнецова Анастасия Натальяовна, Программист, 24.11.2021, 6 дней

Попова Александра Юлияовна, Банкир, 24.11.2021, 6 дней

Фёдорова Василиса Кристинаовна, Декоратор, 16.11.2021, 14 дней

Смирнова Надежда Людмилаовна, Генетик, 20.11.2021, 10 дней

Морозова Юлия Еленаовна, Повар, 11.11.2021, 19 дней

Лебедева Алиса Екатеринаовна, Генетик, 3.11.2021, 27 дней

Петрова Кристина Кристинаовна, Композитор, 16.11.2021, 14 дней

Волкова Наталья Еваовна, Археолог, 29.11.2021, 1 дней

Козлова Ангелина Еваовна, Пожарный, 27.11.2021, 3 дней

Попова Екатерина Любовьовна, Инженер, 4.11.2021, 26 дней

Иванова Елизавета Вераовна, Археолог, 13.11.2021, 17 дней

Попова Кристина Алисаовна, Пожарный, 2.11.2021, 28 дней

Павлова Юлия Анастасияовна, Авиадиспетчер, 10.11.2021, 20 дней

Фёдорова Екатерина Юлияовна, Хирург, 17.11.2021, 13 дней

Воробьёва Александра Елизаветаовна, Пожарный, 29.11.2021, 1 дней

Соловьёва Нина Александраовна, Дизайнер ландшафта, 25.11.2021, 5 дней

Смирнова Людмила Еваовна, Композитор, 19.11.2021, 11 дней

Васильева Елизавета Кираовна, Актёр, 16.11.2021, 14 дней

Волкова Екатерина Анастасияовна, Генетик, 20.11.2021, 10 дней

Петрова Ирина Любовьовна, Банкир, 8.11.2021, 22 дней

Иванова Ольга Анастасияовна, Исследователь, 13.11.2021, 17 дней

Иванова София Иринаовна, Композитор, 21.11.2021, 9 дней

Богданова Кира Юлияовна, Композитор, 6.11.2021, 24 дней

Павлова София Еваовна, Пожарный, 11.11.2021, 19 дней

Иванова Наталья Вераовна, Звукооператор, 14.11.2021, 16 дней

Соловьёва Полина Василисаовна, Дизайнер ландшафта, 16.11.2021, 14 дней

Кузнецова Наталья Софияовна, Исследователь, 16.11.2021, 14 дней

Смирнова Кристина Екатеринаовна, Программист, 26.11.2021, 4 дней

Петрова Любовь Екатеринаовна, Врач, 8.11.2021, 22 дней

Соколова Елена Полинаовна, Хирург, 6.11.2021, 24 дней

Соловьёва Ангелина Натальяовна, Композитор, 9.11.2021, 21 дней

Павлова Екатерина Полинаовна, Дизайнер, 8.11.2021, 22 дней

Зайцева Полина Ольгаовна, Микробиолог, 21.11.2021, 9 дней

Попова Кира Александраовна, Руководитель, 12.11.2021, 18 дней

Васильева Ева Людмилаовна, Системный программист, 23.11.2021, 7 дней

Соловьёва Лидия Еваовна, Продюсер, 14.11.2021, 16 дней

Богданова Ольга Светланаовна, Продюсер, 26.11.2021, 4 дней

Иванова Александра Надеждаовна, Программист, 18.11.2021, 12 дней

Петрова Нина Нинаовна, Актёр, 14.11.2021, 16 дней

Зайцева Юлия Василисаовна, Микробиолог, 9.11.2021, 21 дней

Семёнова София Екатеринаовна, Авиадиспетчер, 10.11.2021, 20 дней

Богданова Надежда Софияовна, Генетик, 7.11.2021, 23 дней

Кузнецова Вера Иринаовна, Переводчик, 27.11.2021, 3 дней

Кузнецова Анастасия Кираовна, Инженер, 5.11.2021, 25 дней

Богданова Любовь Ольгаовна, Космонавт, 2.11.2021, 28 дней

Зайцева Василиса Любовьовна, Исследователь, 27.11.2021, 3 дней

Иванова Юлия Вераовна, Авиадиспетчер, 22.11.2021, 8 дней

Новикова Василиса Викторияовна, Генетик, 16.11.2021, 14 дней

Кузнецова Светлана Лидияовна, Системный программист, 18.11.2021, 12 дней

Воробьёва Вера Светланаовна, Программист, 19.11.2021, 11 дней

Попова Любовь Нинаовна, Программист, 17.11.2021, 13 дней

Козлова Екатерина Надеждаовна, Генетик, 13.11.2021, 17 дней

Виноградова Ева Людмилаовна, Продюсер, 18.11.2021, 12 дней

Волкова София Екатеринаовна, Хирург, 18.11.2021, 12 дней

Богданова Александра Ольгаовна, Астроном, 27.11.2021, 3 дней

Смирнова Кристина Екатеринаовна, Системный программист, 10.11.2021, 20 дней

Иванова Ольга Алисаовна, Дизайнер, 29.11.2021, 1 дней

Волкова Кира Кираовна, Звукооператор, 7.11.2021, 23 дней

Зайцева Ольга Анастасияовна, Актёр, 20.11.2021, 10 дней

Воробьёва Виктория Еваовна, Повар, 29.11.2021, 1 дней

Соколова Людмила Надеждаовна, Повар, 12.11.2021, 18 дней

Козлова Елена Надеждаовна, Астроном, 4.11.2021, 26 дней

Павлова Ольга Викторияовна, Генетик, 6.11.2021, 24 дней

Воробьёва Елена Полинаовна, Переводчик, 4.11.2021, 26 дней

Лебедева Вера Василисаовна, Авиадиспетчер, 20.11.2021, 10 дней

Виноградова Надежда Кираовна, Системный программист, 21.11.2021, 9 дней

Соловьёва Кристина Викторияовна, Астроном, 15.11.2021, 15 дней

Новикова Кристина Надеждаовна, Исследователь, 11.11.2021, 19 дней

Козлова Кира Лидияовна, Археолог, 27.11.2021, 3 дней

Попова Нина Софияовна, Композитор, 22.11.2021, 8 дней

Павлова Ирина Нинаовна, Археолог, 26.11.2021, 4 дней

Козлова Нина Екатеринаовна, Микробиолог, 28.11.2021, 2 дней

Иванова Кристина Светланаовна, Руководитель, 13.11.2021, 17 дней

Лебедева Лидия Иринаовна, Авиадиспетчер, 13.11.2021, 17 дней

Морозова Нина Александраовна, Астроном, 11.11.2021, 19 дней

Попова Юлия Надеждаовна, Композитор, 23.11.2021, 7 дней

Волкова Кира Надеждаовна, Пожарный, 8.11.2021, 22 дней

Павлова Кира Вераовна, Программист, 9.11.2021, 21 дней

Виноградова Ирина Вераовна, Микробиолог, 5.11.2021, 25 дней

Новикова Нина Кираовна, Программист, 15.11.2021, 15 дней

Иванова Ангелина Елизаветаовна, Генетик, 16.11.2021, 14 дней

Смирнова Любовь Натальяовна, Банкир, 2.11.2021, 28 дней

Фёдорова Юлия Полинаовна, Дизайнер ландшафта, 11.11.2021, 19 дней

Смирнова Лидия Елизаветаовна, Инженер, 20.11.2021, 10 дней

Попова Ева Елизаветаовна, Исследователь, 21.11.2021, 9 дней

Волкова Кира Вераовна, Пожарный, 27.11.2021, 3 дней

Семёнова Ангелина Лидияовна, Микробиолог, 29.11.2021, 1 дней

Волкова Кира Кристинаовна, Композитор, 22.11.2021, 8 дней

Зайцева Ева Надеждаовна, Руководитель, 7.11.2021, 23 дней

Смирнова Людмила Ольгаовна, Продюсер, 21.11.2021, 9 дней

Попова Надежда Василисаовна, Микробиолог, 6.11.2021, 24 дней

Новикова Ева Ангелинаовна, IT - менеджер, 22.11.2021, 8 дней

Козлова Анастасия Светланаовна, Микробиолог, 13.11.2021, 17 дней

Лебедева Любовь Кристинаовна, Пожарный, 27.11.2021, 3 дней

Семёнова Людмила Надеждаовна, Авиадиспетчер, 22.11.2021, 8 дней

Павлова Наталья Алисаовна, Авиадиспетчер, 12.11.2021, 18 дней

Васильева Ангелина Анастасияовна, Звукооператор, 20.11.2021, 10 дней

Козлова Кира Ольгаовна, Программист, 9.11.2021, 21 дней

Семёнова Анастасия Елизаветаовна, Исследователь, 22.11.2021, 8 дней

Зайцева Людмила Нинаовна, Повар, 18.11.2021, 12 дней

Волкова Ангелина Нинаовна, Врач, 15.11.2021, 15 дней

Зайцева Екатерина Юлияовна, Переводчик, 8.11.2021, 22 дней

Морозова Лидия Еленаовна, Космонавт, 23.11.2021, 7 дней

Иванова Полина Кираовна, Авиадиспетчер, 8.11.2021, 22 дней

Смирнова Наталья Натальяовна, Пожарный, 15.11.2021, 15 дней

Воробьёва Любовь Еваовна, Руководитель, 11.11.2021, 19 дней

Петрова Лидия Лидияовна, Системный программист, 29.11.2021, 1 дней

Иванова Василиса Надеждаовна, Актёр, 13.11.2021, 17 дней

Смирнова Ангелина Вераовна, Дизайнер ландшафта, 7.11.2021, 23 дней

Соловьёва Ангелина Любовьовна, Переводчик, 19.11.2021, 11 дней

Петрова Ирина Натальяовна, Банкир, 16.11.2021, 14 дней

Козлова Анастасия Любовьовна, Астроном, 12.11.2021, 18 дней

Зайцева Василиса Алисаовна, Повар, 27.11.2021, 3 дней

Попова Алиса Любовьовна, Композитор, 29.11.2021, 1 дней

Новикова Алиса Полинаовна, Декоратор, 19.11.2021, 11 дней

Богданова Людмила Кираовна, Дизайнер ландшафта, 20.11.2021, 10 дней

Иванова Василиса Василисаовна, IT - менеджер, 7.11.2021, 23 дней

Павлова Вера Любовьовна, Врач, 28.11.2021, 2 дней

Козлова Надежда Любовьовна, Дизайнер, 2.11.2021, 28 дней

Семёнова Юлия Надеждаовна, Археолог, 7.11.2021, 23 дней

Воробьёва Кристина Екатеринаовна, Инженер, 29.11.2021, 1 дней

Павлова Елена Надеждаовна, Врач, 10.11.2021, 20 дней

Зайцева Надежда Вераовна, Системный программист, 28.11.2021, 2 дней

Попова Алиса Анастасияовна, IT - менеджер, 4.11.2021, 26 дней

Соколова Светлана Екатеринаовна, Врач, 9.11.2021, 21 дней

Попова Полина Елизаветаовна, Исследователь, 4.11.2021, 26 дней

Смирнова Василиса Иринаовна, Звукооператор, 23.11.2021, 7 дней

Смирнова Людмила Екатеринаовна, Системный программист, 6.11.2021, 24 дней

Волкова Елена Александраовна, Программист, 21.11.2021, 9 дней

Петрова Александра Светланаовна, Авиадиспетчер, 21.11.2021, 9 дней

Попова Полина Екатеринаовна, Космонавт, 17.11.2021, 13 дней

Волкова Виктория Полинаовна, Программист, 9.11.2021, 21 дней

Волкова Вера Алисаовна, Пожарный, 19.11.2021, 11 дней

Павлова Людмила Анастасияовна, Микробиолог, 1.11.2021, 29 дней

Богданова Виктория Ольгаовна, Микробиолог, 11.11.2021, 19 дней

Зайцева Нина Еленаовна, Дизайнер ландшафта, 14.11.2021, 16 дней

Попова Елизавета Кристинаовна, Руководитель, 23.11.2021, 7 дней

Иванова Ангелина Софияовна, Пожарный, 29.11.2021, 1 дней

Козлова Светлана Светланаовна, Врач, 1.11.2021, 29 дней

Богданова Светлана Алисаовна, Актёр, 12.11.2021, 18 дней

Смирнова Лидия Кристинаовна, IT - менеджер, 6.11.2021, 24 дней

Соколова Александра Василисаовна, Продюсер, 29.11.2021, 1 дней

Соловьёва Нина Людмилаовна, Микробиолог, 9.11.2021, 21 дней

Воробьёва Светлана Полинаовна, Переводчик, 19.11.2021, 11 дней

Соколова Светлана Еваовна, Авиадиспетчер, 3.11.2021, 27 дней

Попова Вера Полинаовна, Декоратор, 13.11.2021, 17 дней

Павлова Ева Юлияовна, Инженер, 21.11.2021, 9 дней

Петрова Ирина Нинаовна, Звукооператор, 3.11.2021, 27 дней

Васильева Ольга Надеждаовна, Дизайнер, 25.11.2021, 5 дней

Васильева Кира Алисаовна, Переводчик, 22.11.2021, 8 дней

Соколова Наталья Александраовна, Дизайнер, 21.11.2021, 9 дней

Новикова Вера Полинаовна, Системный программист, 20.11.2021, 10 дней

Зайцева Александра Полинаовна, Дизайнер ландшафта, 22.11.2021, 8 дней

Волкова София Людмилаовна, Банкир, 14.11.2021, 16 дней

Соловьёва Елизавета Кираовна, Продюсер, 19.11.2021, 11 дней

Попова Любовь Вераовна, Звукооператор, 27.11.2021, 3 дней

Петрова Надежда Кристинаовна, Микробиолог, 20.11.2021, 10 дней

Павлова Александра Ольгаовна, Исследователь, 3.11.2021, 27 дней

Соколова Лидия Светланаовна, Программист, 10.11.2021, 20 дней

Волкова Светлана Еваовна, Генетик, 15.11.2021, 15 дней

Соловьёва Вера Светланаовна, Дизайнер ландшафта, 19.11.2021, 11 дней

Лебедева Светлана Ольгаовна, Дизайнер, 20.11.2021, 10 дней

Волкова Юлия Ангелинаовна, Системный программист, 27.11.2021, 3 дней

Богданова Ирина Ольгаовна, Повар, 17.11.2021, 13 дней

Соколова Василиса Еленаовна, Пожарный, 14.11.2021, 16 дней

Соловьёва Полина Кристинаовна, Продюсер, 17.11.2021, 13 дней

Кузнецова Ева Софияовна, IT - менеджер, 22.11.2021, 8 дней

Морозова Кира Людмилаовна, Дизайнер, 25.11.2021, 5 дней

Кузнецова Ирина Еваовна, Звукооператор, 8.11.2021, 22 дней

Соловьёва Юлия Анастасияовна, Исследователь, 5.11.2021, 25 дней

Кузнецова Наталья Полинаовна, Астроном, 23.11.2021, 7 дней

Васильева Нина Полинаовна, Хирург, 7.11.2021, 23 дней

Виноградова Анастасия Людмилаовна, Хирург, 14.11.2021, 16 дней

Волкова Виктория Еленаовна, Системный программист, 10.11.2021, 20 дней

# Пример выходного файла (на 200 записей):

Количество записей: 200

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 285 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 56 милисекунд

Результат сортировки (отсортированные данные):

[ IT - менеджер, 4, Вера, Попова, Полинаовна]

[ IT - менеджер, 8, Ева, Кузнецова, Софияовна]

[ IT - менеджер, 8, Ева, Новикова, Ангелинаовна]

[ IT - менеджер, 15, Ольга, Воробьёва, Викторияовна]

[ IT - менеджер, 23, Василиса, Иванова, Василисаовна]

[ IT - менеджер, 24, Лидия, Васильева, Натальяовна]

[ IT - менеджер, 24, Лидия, Смирнова, Кристинаовна]

[ IT - менеджер, 25, Кристина, Иванова, Натальяовна]

[ IT - менеджер, 26, Алиса, Попова, Анастасияовна]

[ Авиадиспетчер, 8, Людмила, Семёнова, Надеждаовна]

[ Авиадиспетчер, 8, Юлия, Иванова, Вераовна]

[ Авиадиспетчер, 9, Александра, Петрова, Светланаовна]

[ Авиадиспетчер, 10, Вера, Лебедева, Василисаовна]

[ Авиадиспетчер, 17, Лидия, Лебедева, Иринаовна]

[ Авиадиспетчер, 18, Наталья, Павлова, Алисаовна]

[ Авиадиспетчер, 20, София, Семёнова, Екатеринаовна]

[ Авиадиспетчер, 20, Юлия, Павлова, Анастасияовна]

[ Авиадиспетчер, 22, Полина, Иванова, Кираовна]

[ Авиадиспетчер, 24, Елена, Иванова, Анастасияовна]

[ Авиадиспетчер, 27, Светлана, Соколова, Еваовна]

[ Авиадиспетчер, 28, Елизавета, Козлова, Василисаовна]

[ Актёр, 10, Ольга, Зайцева, Анастасияовна]

[ Актёр, 14, Елизавета, Васильева, Кираовна]

[ Актёр, 16, Нина, Петрова, Нинаовна]

[ Актёр, 17, Василиса, Иванова, Надеждаовна]

[ Актёр, 18, Светлана, Богданова, Алисаовна]

[ Актёр, 25, Александра, Виноградова, Елизаветаовна]

[ Археолог, 1, Наталья, Волкова, Еваовна]

[ Археолог, 3, Кира, Козлова, Лидияовна]

[ Археолог, 4, Ирина, Павлова, Нинаовна]

[ Археолог, 4, Полина, Воробьёва, Полинаовна]

[ Археолог, 17, Елизавета, Иванова, Вераовна]

[ Археолог, 23, Юлия, Семёнова, Надеждаовна]

[ Астроном, 3, Александра, Богданова, Ольгаовна]

[ Астроном, 7, Наталья, Кузнецова, Полинаовна]

[ Астроном, 15, Кристина, Соловьёва, Викторияовна]

[ Астроном, 18, Анастасия, Козлова, Любовьовна]

[ Астроном, 19, Нина, Морозова, Александраовна]

[ Астроном, 26, Елена, Козлова, Надеждаовна]

[ Банкир, 2, Елена, Виноградова, Екатеринаовна]

[ Банкир, 6, Александра, Попова, Юлияовна]

[ Банкир, 14, Ирина, Петрова, Натальяовна]

[ Банкир, 16, София, Волкова, Людмилаовна]

[ Банкир, 20, Ольга, Попова, Юлияовна]

[ Банкир, 22, Ирина, Петрова, Любовьовна]

[ Банкир, 24, Нина, Петрова, Еленаовна]

[ Банкир, 28, Любовь, Смирнова, Натальяовна]

[ Врач, 2, Вера, Павлова, Любовьовна]

[ Врач, 15, Ангелина, Волкова, Нинаовна]

[ Врач, 20, Елена, Павлова, Надеждаовна]

[ Врач, 21, Светлана, Соколова, Екатеринаовна]

[ Врач, 22, Любовь, Петрова, Екатеринаовна]

[ Врач, 22, Людмила, Попова, Надеждаовна]

[ Врач, 29, Светлана, Козлова, Светланаовна]

[ Генетик, 10, Екатерина, Волкова, Анастасияовна]

[ Генетик, 10, Надежда, Смирнова, Людмилаовна]

[ Генетик, 14, Ангелина, Иванова, Елизаветаовна]

[ Генетик, 14, Василиса, Новикова, Викторияовна]

[ Генетик, 15, Светлана, Волкова, Еваовна]

[ Генетик, 17, Екатерина, Козлова, Надеждаовна]

[ Генетик, 23, Надежда, Богданова, Софияовна]

[ Генетик, 24, Наталья, Попова, Юлияовна]

[ Генетик, 24, Ольга, Павлова, Викторияовна]

[ Генетик, 27, Алиса, Лебедева, Екатеринаовна]

[ Декоратор, 11, Алиса, Новикова, Полинаовна]

[ Декоратор, 14, Василиса, Фёдорова, Кристинаовна]

[ Декоратор, 17, Вера, Попова, Полинаовна]

[ Дизайнер, 1, Ольга, Иванова, Алисаовна]

[ Дизайнер, 5, Кира, Морозова, Людмилаовна]

[ Дизайнер, 5, Ольга, Васильева, Надеждаовна]

[ Дизайнер, 9, Наталья, Соколова, Александраовна]

[ Дизайнер, 10, Светлана, Лебедева, Ольгаовна]

[ Дизайнер, 22, Екатерина, Павлова, Полинаовна]

[ Дизайнер, 28, Надежда, Козлова, Любовьовна]

[ Дизайнер ландшафта, 5, Нина, Соловьёва, Александраовна]

[ Дизайнер ландшафта, 8, Александра, Зайцева, Полинаовна]

[ Дизайнер ландшафта, 10, Людмила, Богданова, Кираовна]

[ Дизайнер ландшафта, 11, Вера, Соловьёва, Светланаовна]

[ Дизайнер ландшафта, 14, Полина, Соловьёва, Василисаовна]

[ Дизайнер ландшафта, 16, Нина, Зайцева, Еленаовна]

[ Дизайнер ландшафта, 19, Юлия, Фёдорова, Полинаовна]

[ Дизайнер ландшафта, 23, Ангелина, Смирнова, Вераовна]

[ Дизайнер ландшафта, 24, София, Лебедева, Людмилаовна]

[ Звукооператор, 3, Любовь, Попова, Вераовна]

[ Звукооператор, 7, Василиса, Смирнова, Иринаовна]

[ Звукооператор, 10, Ангелина, Васильева, Анастасияовна]

[ Звукооператор, 13, Ангелина, Воробьёва, Александраовна]

[ Звукооператор, 16, Наталья, Иванова, Вераовна]

[ Звукооператор, 22, Ирина, Кузнецова, Еваовна]

[ Звукооператор, 23, Кира, Волкова, Кираовна]

[ Звукооператор, 27, Ирина, Петрова, Нинаовна]

[ Инженер, 1, Кристина, Воробьёва, Екатеринаовна]

[ Инженер, 9, Ева, Павлова, Юлияовна]

[ Инженер, 10, Лидия, Смирнова, Елизаветаовна]

[ Инженер, 12, Полина, Фёдорова, Натальяовна]

[ Инженер, 25, Анастасия, Кузнецова, Кираовна]

[ Инженер, 26, Екатерина, Попова, Любовьовна]

[ Исследователь, 3, Василиса, Зайцева, Любовьовна]

[ Исследователь, 8, Анастасия, Семёнова, Елизаветаовна]

[ Исследователь, 9, Ева, Попова, Елизаветаовна]

[ Исследователь, 14, Наталья, Кузнецова, Софияовна]

[ Исследователь, 17, Ольга, Иванова, Анастасияовна]

[ Исследователь, 19, Кристина, Новикова, Надеждаовна]

[ Исследователь, 25, Юлия, Соловьёва, Анастасияовна]

[ Исследователь, 26, Полина, Попова, Елизаветаовна]

[ Исследователь, 27, Александра, Павлова, Ольгаовна]

[ Композитор, 1, Алиса, Попова, Любовьовна]

[ Композитор, 7, Юлия, Попова, Надеждаовна]

[ Композитор, 8, Кира, Волкова, Алисаовна]

[ Композитор, 8, Кира, Волкова, Кристинаовна]

[ Композитор, 8, Нина, Попова, Софияовна]

[ Композитор, 9, София, Иванова, Иринаовна]

[ Композитор, 11, Людмила, Смирнова, Еваовна]

[ Композитор, 14, Кристина, Петрова, Кристинаовна]

[ Композитор, 21, Ангелина, Соловьёва, Натальяовна]

[ Композитор, 24, Кира, Богданова, Юлияовна]

[ Космонавт, 7, Лидия, Морозова, Еленаовна]

[ Космонавт, 13, Полина, Попова, Екатеринаовна]

[ Космонавт, 28, Любовь, Богданова, Ольгаовна]

[ Микробиолог, 1, Ангелина, Семёнова, Лидияовна]

[ Микробиолог, 2, Нина, Козлова, Екатеринаовна]

[ Микробиолог, 9, Полина, Зайцева, Ольгаовна]

[ Микробиолог, 10, Ангелина, Попова, Натальяовна]

[ Микробиолог, 10, Надежда, Петрова, Кристинаовна]

[ Микробиолог, 17, Анастасия, Козлова, Светланаовна]

[ Микробиолог, 19, Виктория, Богданова, Ольгаовна]

[ Микробиолог, 21, Нина, Соловьёва, Людмилаовна]

[ Микробиолог, 21, Юлия, Зайцева, Василисаовна]

[ Микробиолог, 24, Надежда, Попова, Василисаовна]

[ Микробиолог, 25, Ирина, Виноградова, Вераовна]

[ Микробиолог, 29, Людмила, Павлова, Анастасияовна]

[ Переводчик, 3, Вера, Кузнецова, Иринаовна]

[ Переводчик, 8, Кира, Васильева, Алисаовна]

[ Переводчик, 11, Ангелина, Соловьёва, Любовьовна]

[ Переводчик, 11, Светлана, Воробьёва, Полинаовна]

[ Переводчик, 22, Екатерина, Зайцева, Юлияовна]

[ Переводчик, 26, Елена, Воробьёва, Полинаовна]

[ Повар, 1, Виктория, Воробьёва, Еваовна]

[ Повар, 3, Василиса, Зайцева, Алисаовна]

[ Повар, 8, Ирина, Соколова, Людмилаовна]

[ Повар, 12, Людмила, Зайцева, Нинаовна]

[ Повар, 13, Ирина, Богданова, Ольгаовна]

[ Повар, 17, Надежда, Петрова, Екатеринаовна]

[ Повар, 18, Людмила, Соколова, Надеждаовна]

[ Повар, 19, Юлия, Морозова, Еленаовна]

[ Пожарный, 1, Александра, Воробьёва, Елизаветаовна]

[ Пожарный, 1, Ангелина, Иванова, Софияовна]

[ Пожарный, 1, Ангелина, Лебедева, Лидияовна]

[ Пожарный, 3, Ангелина, Козлова, Еваовна]

[ Пожарный, 3, Кира, Волкова, Вераовна]

[ Пожарный, 3, Любовь, Лебедева, Кристинаовна]

[ Пожарный, 11, Вера, Волкова, Алисаовна]

[ Пожарный, 15, Наталья, Смирнова, Натальяовна]

[ Пожарный, 16, Василиса, Соколова, Еленаовна]

[ Пожарный, 19, София, Павлова, Еваовна]

[ Пожарный, 22, Кира, Волкова, Надеждаовна]

[ Пожарный, 28, Кристина, Попова, Алисаовна]

[ Программист, 4, Кристина, Смирнова, Екатеринаовна]

[ Программист, 6, Анастасия, Кузнецова, Натальяовна]

[ Программист, 9, Елена, Волкова, Александраовна]

[ Программист, 11, Вера, Воробьёва, Светланаовна]

[ Программист, 12, Александра, Иванова, Надеждаовна]

[ Программист, 13, Любовь, Попова, Нинаовна]

[ Программист, 15, Нина, Новикова, Кираовна]

[ Программист, 20, Лидия, Соколова, Светланаовна]

[ Программист, 21, Виктория, Волкова, Полинаовна]

[ Программист, 21, Кира, Козлова, Ольгаовна]

[ Программист, 21, Кира, Павлова, Вераовна]

[ Продюсер, 1, Александра, Соколова, Василисаовна]

[ Продюсер, 4, Ольга, Богданова, Светланаовна]

[ Продюсер, 9, Людмила, Смирнова, Ольгаовна]

[ Продюсер, 11, Елизавета, Соловьёва, Кираовна]

[ Продюсер, 12, Ева, Виноградова, Людмилаовна]

[ Продюсер, 13, Полина, Соловьёва, Кристинаовна]

[ Продюсер, 16, Лидия, Соловьёва, Еваовна]

[ Продюсер, 21, Светлана, Лебедева, Светланаовна]

[ Продюсер, 22, Надежда, Иванова, Викторияовна]

[ Руководитель, 7, Елизавета, Попова, Кристинаовна]

[ Руководитель, 12, Александра, Соколова, Вераовна]

[ Руководитель, 17, Кристина, Иванова, Светланаовна]

[ Руководитель, 18, Кира, Попова, Александраовна]

[ Руководитель, 19, Любовь, Воробьёва, Еваовна]

[ Руководитель, 23, Ева, Зайцева, Надеждаовна]

[ Системный программист, 1, Лидия, Петрова, Лидияовна]

[ Системный программист, 2, Надежда, Зайцева, Вераовна]

[ Системный программист, 3, Юлия, Волкова, Ангелинаовна]

[ Системный программист, 7, Ева, Васильева, Людмилаовна]

[ Системный программист, 9, Надежда, Виноградова, Кираовна]

[ Системный программист, 10, Вера, Новикова, Полинаовна]

[ Системный программист, 12, Светлана, Кузнецова, Лидияовна]

[ Системный программист, 20, Виктория, Волкова, Еленаовна]

[ Системный программист, 20, Кристина, Смирнова, Екатеринаовна]

[ Системный программист, 24, Людмила, Смирнова, Екатеринаовна]

[ Хирург, 3, София, Фёдорова, Елизаветаовна]

[ Хирург, 8, Людмила, Козлова, Вераовна]

[ Хирург, 12, София, Волкова, Екатеринаовна]

[ Хирург, 13, Екатерина, Фёдорова, Юлияовна]

[ Хирург, 16, Анастасия, Виноградова, Людмилаовна]

[ Хирург, 23, Нина, Васильева, Полинаовна]

[ Хирург, 24, Елена, Соколова, Полинаовна]

# Тесты

Рассмотрим 3 случая:

Мы будем автоматически генерировать случайные данные.

## 1. 20 случайно сгенерированных записей:

Количество записей: 20

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 4 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 2 милисекунд

Быстрая сортировка на 50% быстрее Двухпутевой

## 2. 200 случайно сгенерированных записей:

Количество записей: 200

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 254 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 44 милисекунд

Быстрая сортировка на 82% быстрее Двухпутевой

## 3. 2000 случайно сгенерированных записей:

Количество записей: 2000

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 14097 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 522 милисекунд

Быстрая сортировка на 96% быстрее Двухпутевой

## Результат:

Я думаю рассматривать большее количество записей не имеет смысла, мы уже можем огласить результаты:

При случайно сгенерированных данных быстрая сортировка в среднем работает на 76% быстрее, чем сортировка Двухпутевыми вставками.

Чем больше данных, тем быстрее сортирует Быстрая сортировка, и тем медленнее сортирует Двухпутевая сортировка.

# Примеры, на которых достигается худший и лучший случаи

Рассмотрим 3 случая входных данных:

## 1. Входные данные уже отсортированы:

Количество записей: 100

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 6 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 21 милисекунд

## 2. Входные данные отсортированы в обратном порядке:

Количество записей: 100

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 11 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 26 милисекунд

## 3. Входные данные содержат 100 одинаковых записей:

Количество записей: 100

Сортировка Двухпутевыми вставками:

Время, затраченное на сортировку: 9 милисекунд

Сортировка Быстрой сортировкой:

Время, затраченное на сортировку: 27 милисекунд

## Результат:

Сортировка Двухпутевыми вставками сортирует уже упорядоченные данные, в среднем на 64% быстрее, чем Быстрая сортировка.

# Проверка устойчивости с обоснованием результатов

Введём новую структуру записи, содержащей поле 'индекс'

Заполним массив из 2х записей одинаковыми значениями, с разными индексами

Отсортируем 2мя разными способами и проверим, устойчива ли каждая сортировка или нет :

Выводим данные без изменений :

[Учитель, 10, Крестникова, Ольга, Александровна] , ind = 0

[Учитель, 10, Крестникова, Ольга, Александровна], ind = 1

Сортируем Двухпутевыми вставками

Выводим данные отсортированные Двухпутевыми вставками :

[Учитель, 10, Крестникова, Ольга, Александровна] , ind = 0

[Учитель, 10, Крестникова, Ольга, Александровна], ind = 1

Сортируем Быстрой сортировкой

Выводим данные отсортированные Быстрой сортировкой :

[Учитель, 10, Крестникова, Ольга, Александровна] , ind = 1

[Учитель, 10, Крестникова, Ольга, Александровна], ind = 0

## Результат:

Как мы видим, Двухпутевая сортировка является устойчивой, а Быстрая сортировка является неустойчивой.

# Выводы:

Сортировка Двухпутевыми вставками работает быстрее тогда, когда нужно отсортировать небольшое количество записей. Также эта сортировка показывает хорошую производительность, когда нужно отсортировать уже упорядоченные записи.

Быстрая сортировка работает примерно одинаково, с любым количеством элементов. Также она показывает большую производительность, когда нужно отсортировать случайные записи, но показывает достаточно низкую производительность, когда нужно отсортировать уже отсортированные записи.