Федеральное агенство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

"Сибирский государственный университет телекоммуникаций и информатики"

Лабораторная работа №5

по дисциплине "Структуры и алгоритмы обработки данных"

Выполнил: студент группы:

Проверил: ассистент кафедры ПМиК

Дьячкова И.С.

Новосибирск 2020

**Содержание:**

1. **Цель работы**
2. **Описание алгоритмов**
3. **Листинг программы**
4. **Результат работы программы**

**1.Цель работы:**

1). Разработать процедуру сортировки массива целых чисел методом прямого включения (InsertSort).

2). Предусмотреть подсчет фактического количества пересылок и сравнений (Мф и Сф), сравнить с теоретическими оценками М и С.

3). Сравнить время работы методов квадратичной трудоемкости на массивах случайных чисел. Построить таблицу вида:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Select | Bubble | Shaker | Insert |
| 100 | Мф+Сф |  |  |  |
| 200 |  |  |  |  |
| 300 |  |  |  |  |
| 400 |  |  |  |  |
| 500 |  |  |  |  |

3). Дополнительное задание (на 5+):

Построить на экране в одной координатной плоскости графики зависимости трудоемкости (Мф+Сф) от n для всех методов квадратичной трудоемкости (для массива случайных чисел).

**2.Описание алгоритмов**

**void Fillinc(int A[N])**

Заполняет массив А возрастающими числами

**void FillDec(int A[N])**

Заполняет массив А убывающими числами.

**void FillRand(int A[N])**

Заполняет массив А случайными числами.

**void CheckSum(int A[N])**

Считает контрольную сумму элементов массива А.

**void RunNumber(int A[N])**

Считает серии в массиве А.

**void PrintMas(int A[N])**

Выводит на экран элементы массива А.

**void SelectSort (int A[], int n)**

Сортировки массива методом прямого выбора

**void BubbleSort (int A[], int n)**

Сортировки массива пузырьковым методом

**void ShakerSort(int A[], int n)**

Шейкерная сортировка

**void InsertSort (int A[], int n)**

Сортировка массива методом прямого включения

**3. Листинг программы**

#include <iostream>

#include <stdlib.h>

#include <stdio.h>

#include <graphics.h>

#include <time.h>

using namespace std;

int i=0, M = 0, C = 0;

int random (int N) {

return rand() % N;

}

void FillInc (int A[], int n) {

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

A[i] = i;

}

}

void FillDec (int A[], int n) {

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

A[i] = n-i;

}

}

void FillRand(int A[], int n) {

int i;

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

A[i] = 0;

}

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

A[i] = rand()% 10;

}

}

void PrintMas (int A[], int n) {

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

printf("%d ", A[i]);

}

}

void InsertSort (int A[], int n) {

int t=0, j=0;

C=0; M=0;

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

t=A[i];

j=i-1;

M++;

C++;

if ( j>-1 && t<A[j]) C--;

while (j>=0 && t<A[j]){

A[j+1]=A[j];

M++;

j=j-1;

C++;

}

A[j+1]=t;

M++;

}

}

void SelectSort (int A[], int n) {

int j, min, temp;

M=0; C=0;

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

min = i;

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

C += 1;

if (A[j] < A[min]) {

min = j;

}

}

temp = A[i];

A[i] = A[min];

A[min] = temp;

M += 3;

}

}

void BubbleSort (int A[], int n) {

int j, min, temp;

M=0; C=0;

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

for (j=n-2; j>=i; j--) {

C += 1;

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

M += 3;

temp = A[j];

A[j] = A[j+1];

A[j+1] = temp;

}

}

}

}

void ShakerSort(int A[], int n)

{

int Left=0, Right=n-1, k=n ,temp=0;

do

{

for (i=Right; i>=Left+1; i--){

if (A[i]<A[i-1]) {

M+=3;

temp=A[i];

A[i]=A[i-1];

A[i-1]=temp;k=i;}C+=1;}

Left=k;

for (i=Left; i<=Right-1; i++){

if (A[i]>A[i+1]) {

M+=3;

temp=A[i];

A[i]=A[i+1];

A[i+1]=temp;

k=i;

} C+=1;}

Right=k;

} while (Left<Right);

}

int main () {

using namespace std;

int Sum,ser,n,y;

int w;

setlocale(LC\_ALL,"RUS");

srand(time(NULL));

n=10;

int A[n];

printf("Убывающий массив: ");

FillDec(A, n);

PrintMas(A,n);

printf("\n ");

printf("\n ");

printf("Отсортированный массив: ");

InsertSort(A, n);

PrintMas(A,n);

printf("\n ");

printf("M=%d C=%d", M, C);

printf("\n ");

printf("M+C=%d", M+C);

printf("\n\n");

printf(" Select Bubble Shaker Insert Возр. Убыв.\n");

for(n=100;n<501;n+=100) {

int B[n], P[n], V[n], U[n], S[n], Z[n];

C=0;M=0;

printf("n=%d",n);

FillRand(B,n);

SelectSort(B,n);

printf(" %d",M+C);

C=0;M=0;

FillRand(P,n);

BubbleSort(P,n);

printf(" %d",M+C);

C=0;M=0;

FillRand(V,n);

ShakerSort(V,n);

printf(" %d",M+C);

C=0;M=0;

FillRand(U,n);

InsertSort(U,n);

printf(" %d",M+C);

C=0;M=0;

FillInc(S,n);

InsertSort(S,n);

printf(" %d",M+C);

C=0;M=0;

FillDec(Z,n);

InsertSort(Z,n);

printf(" %d",M+C);

C=0;M=0;

printf("\n\n");

}

system("PAUSE");

initwindow(700, 700);

line(50,50,50,650);

line(650,650,50,650);

moveto(50, 650);

M=0; C=0;

setcolor(2); //зелёный

for(n=2;n<300;n++){

int E[n];

FillRand(E,n);

SelectSort(E,n);

lineto(n\*20,650-(M+C));

}

moveto(50, 650);

setcolor(4); //красный

M=0; C=0;

for(n=2;n<300;n++){

int Z[n];

FillRand(Z,n);

BubbleSort(Z,n);

lineto(n\*20,650-(M+C));

}

moveto(50, 650);

setcolor(5); //фиолетовый

M=0; C=0;

for(n=2;n<300;n++){

int Q[n];

FillRand(Q,n);

ShakerSort(Q,n);

lineto(n\*20,650-(M+C));

}

moveto(50, 650);

setcolor(6); //коричневый

M=0; C=0;

for(n=2;n<300;n++){

int G[n];

FillRand(G,n);

InsertSort(G,n);

lineto(n\*20,650-(M+C));

}

getch();

closegraph();

return 0;

}

**4. Результат работы программы**



