**Лабораторная работа №15. Алгоритмы сортировки**

**Вариант 10**

Условие:

Определить зависимость времени выполнения алгоритмов от количества элементов для каждого из алгоритмов. Выполнить моделирование для массивов размером 1000, 2000, 3000, 4000, 5000. Произвести сравнение эффективности алгоритмов (построить график в приложении Excel).

Сортировка разделением, сортировка выбором, сортировка Шелла

Код:

Главная функция:

#include "stdafx.h"

#include <ctime>

#include<time.h>

#include <iostream>

using namespace std;

void BubbleSort1(int \*A, int N)

{ int i, j, count, key;

for (i = 0; i < N; i++)

{ for (j = 0; j < N - 1; j++)

{ key = j + 1; count = A[key];

if (A[j] > A[key])

{ A[key] = A[j];

A[j] = count; }

}

}

/\* cout<<"Результирующий массив: ";

for (i = 0; i < N; i++) cout<<A[i]<<" "; \*/

}

void SelectSort2(int \*A, int n)

{ int k, i, j;

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

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

if (A[j] < A[k])

k = j;

int c = A[k];

A[k] = A[i];

A[i] = c;

}

/\*cout<<"Результирующий массив: ";

for (i = 0; i < n; i++) cout<<A[i]<<" ";

cout<<endl;\*/

}

void Shell(int \*A, int n) //сортировка Шелла

{ int i, j, c, d = n;

d = d / 2;

while (d > 0)

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

{ j = i;

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

{ c = A[j];

A[j] = A[j + d];

A[j + d] = c;

j--;

}

}

d = d / 2;

}

/\*cout<<"Результирующий массив: ";

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

cout<<A[i]<<" ";

cout<<endl;\*/

}

int GetHoarBorder(int \*A, int sm, int em)

{

int i = sm - 1, j = em + 1;

int brd = A[sm];

int buf;

while (i < j)

{ while(A[--j]> brd) ;

while(A[++i]< brd) ;

if (i < j)

{ buf = A[j]; A[j] = A[i]; A[i] = buf; };

}

return j;

}

int\* SortHoar(int \*A, int sm, int em)

{ if (sm < em)

{ int hb = GetHoarBorder(A, sm, em);

SortHoar(A, sm, hb);

SortHoar(A, hb + 1,em);

}

return A;

};

void main()

{ setlocale(LC\_ALL, "Rus");

int N; clock\_t t1,t2;

int chisl;

cout<<"Количество элементов > ";

cin>>N;

int \*A=new int[N], \*A1=new int[N],\*B=new int[N];

srand((unsigned)time(NULL));

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

{

A[i]=rand()%2000;

}

cout<<endl;

do{

cout<<"\tМеню:\n";

cout<<"1-Сортировка пузырьком;\n";

cout<<"2-Сортировка выбором\n";

cout<<"3-Сорировка Шелла\n";

cout<<"4-Сортировка Хоара\n";

cout<<"0-Выход;\n";

cin>>chisl;

switch(chisl)

{

case 1: {t1=clock();

BubbleSort1(A,N);

t2=clock();

cout<<endl;

cout<<"Прошло секунд "<<(float(t2-t1))/CLOCKS\_PER\_SEC<<endl;

break;}

case 2: {t1=clock();

SelectSort2(A,N);

t2=clock();

cout<<endl;

cout<<"Прошло секунд "<<(float(t2-t1))/CLOCKS\_PER\_SEC<<endl;

break;}

case 3: {t1=clock();

Shell(A,N);

cout<<endl;

t2=clock();

cout<<"Прошло секунд "<<(float(t2-t1))/CLOCKS\_PER\_SEC<<endl;

break;}

case 4:{t1=clock();

B=SortHoar(A,0,N-1);

t2=clock();

cout<<endl;

cout<<"Прошло секунд "<<(float(t2-t1))/CLOCKS\_PER\_SEC<<endl;

/\*for(int i=0;i<N;i++)

cout<<\*(B+i)<<" ";

cout<<endl;\*/

break;}

case 0: exit(0);

}

}while(chisl!=0);

delete []A;

delete []A1;

delete []B;

}