ALGORITMO DE ORDENAMIENTO BURBUJA

algoritmo burbuja(a, n)

para i🡨0 hasta i<n hacer

para j🡨0 hasta j<(n-1 ) hacer

si a[ j]> a[ j+1 ] entonces

aux🡨a[ j ]

a[ j ]🡨a[ j+1 ]

a[ j +1]🡨aux

finsi

finpara

finpara

finburbuja

PROGRAMA DE ORDENAMIENTO BURBUJA EN LISTAS

#include "iostream.h"

#include "conio.h"

class LISTA{

int a;

LISTA \*sgte,\*raiz;

int cantidad;

public:

LISTA(){ raiz=NULL; cantidad=0; }

void Ing(int x){

LISTA \*n=new LISTA;

n->a=x;

if(raiz==NULL){ raiz=n; n->sgte=NULL; }

else {

n->sgte=raiz;

raiz=n;

}

cantidad++;

}

void VisLista(){

LISTA \*q=raiz;

cout<<"\n";

while(q!=NULL){

cout<<" "<<q->a;

q=q->sgte;

}

}

void buscar(int p, int &dato){

LISTA \*q=raiz;

for(int k=0; k<p; k++){

q=q->sgte;

}

dato=q->a;

}

void colocar(int p, int dato){

LISTA \*q=raiz;

for(int k=0; k<p; k++){

q=q->sgte;

}

q->a=dato;

}

void ordenar(){

int y,x;

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

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

buscar(j, x);

buscar(j+1, y);

if(x<y){ colocar(j, y);

colocar(j+1, x);

}

}

}

};

int main(){

char op;

int a;

LISTA list;

for(;;){

cout<<"\n Ing <1>\n Mos <2>\n Ordenar <3>\n Salir <4>";

op=getch();

switch(op){

case '1':cout<<"\n leer numero: ";cin>>a;

list.Ing(a);

break;

case '2':list.VisLista();break;

case '3':list.ordenar();break;

case '4':return 0;

}

}

}

PROGRAMA DE ORDENAMIENTO BURBUJA EN FILAS

#include "iostream.h"

#include "conio.h"

#include "fstream.h"

#include "io.h"

#include "fcntl.h"

class ENTERO{

int numero;

public:

void ingresar();

void mostrar();

void clasificar();

};

void ENTERO::ingresar(){

ENTERO a;

cout<<"\n\n Ing. numero : ";cin>>a.numero;

ofstream esc("fila",ios::app);

esc.write(reinterpret\_cast<char \*>(&a),sizeof(ENTERO));

esc.close();

}

void ENTERO::mostrar(){ ENTERO a;

ifstream lec("fila");

if(!lec){ cout<<"\n error de archivo "; getch(); }

else {

cout<<"\n\n numero";

lec.read(reinterpret\_cast<char \*>(&a),sizeof(ENTERO));

while(!lec.eof()){

cout<<"\n "<<a.numero;

lec.read(reinterpret\_cast<char \*>(&a),sizeof(ENTERO));

}

getch();

lec.close();

}

}

void ENTERO::clasificar(){

ENTERO a,b;

int i,j;

int h=open("fila",O\_RDWR);

int cantidad= filelength(h)/sizeof(ENTERO);

close(h);

fstream clasif("fila",ios::in|ios::out);

if(!clasif){ cout<<"error al abrir ";getch();}

else{

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

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

clasif.seekg(sizeof(ENTERO)\*j,ios::beg);

clasif.read(reinterpret\_cast<char \*>(&a),sizeof(ENTERO));

clasif.seekg(sizeof(ENTERO)\*(j+1),ios::beg);

clasif.read(reinterpret\_cast<char \*>(&b),sizeof(ENTERO));

if(a.numero > b.numero){

clasif.seekg(sizeof(ENTERO)\*j,ios::beg);

clasif.write(reinterpret\_cast<char \*>(&b),sizeof(ENTERO));

clasif.seekg(sizeof(ENTERO)\*(j+1),ios::beg);

clasif.write(reinterpret\_cast<char \*>(&a),sizeof(ENTERO));

} //fin de if

}

clasif.close();

}

}

int main(){ ENTERO a; char op;

for(;;){

cout<<"\n Ingresar <1>";

cout<<"\n Mostrar <2>";

cout<<"\n clasificar <3>";

cout<<"\n Salir <4>";

op=getch();

switch(op){

case '1':a.ingresar();break;

case '2':a.mostrar();break;

case '3':a.clasificar();break;

case '4': return 0;

}

}

return 0;

}