Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo de Merge

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 03/09/22

Grupo:5D1

Registro: 20300701

**Descripción del programa:**

Se va a crear un programa que ordene un conjunto de números enteros usando el método de Selección, los datos se los dará el usuario y este mismo decidirá hasta cuantos ingresar dentro de un rango de 1 a 50. Este programa no tiene que mostrar la iteraciones, está hecho para acomodar grandes cantidades de datos y de una manera muy eficiente, ya que arma dos mitades o grupos para así acomodarlos por partes cada mitad y así ordenarlas

**Requerimientos:**

a) Ordenará un conjunto de números enteros, la cantidad mínima deberá ser 1 y el máximo 50.

b) Deberá ingresar los valores correspondientes.

c) El o los módulos de ordenación deberán ser métodos.

d) El sistema está diseñado en C++.

e) La ordenación será de menor a mayor.

f) Mostrar el mensaje de error correspondiente a cada posibilidad.

g) Evite usar variables globales.

h) Responda lo siguiente y anexe sus respuestas al reporte de MERGE, estas se revisarán como si fuera un caso 5.

a. ¿Cómo realiza el proceso de división de arreglos?

Con un ciclo for para asignar el rango de la resta de la mitad el valor inicial mas uno para los elementos de la izquierda

Algo como 5-1

Dentro del for es la posición del inicio y se le suma i

Para los de la derecha son los elementos final menos los de la mitad

10-5

Dentro del for es de la mitad mas uno y mas J que incia en 0

b. ¿Cómo realiza el proceso de fusión de arreglos?

Por medio de un ciclo while que en el momento en que un contador nuestro llegue a ser menor al de elementos de el arreglo de la izquierda y el contador de la derecha sea menor al de nuestros elementos de la derecha empiece a fusionar esos valores para incluirlos en nuestro arreglo principal

**PSP:**

|  |  |  |  |
| --- | --- | --- | --- |
| Contenido del PSP – Merge | | | |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 30 | 35 | 17% |
| Tiempo de Codificación. | 90 | 60 | 33% |
| Tiempo estimado de fabricación | 100 | 95 | 5% |
| Total, de líneas de código nuevas. | 50 | 51 | 2% |
| Total, de líneas de código reusadas. | 70 | 48 | 31% |
| Total, de líneas de código del programa. | 130 | 99 | 24% |
| Cantidad de errores de compilación. | 20 | 17 | 15% |
| Cantidad de errores de ejecución. | 15 | 25 | 67% |

**Código fuente:**

// merge.cpp : Este archivo contiene la función "main". La ejecución del programa comienza y termina ahí.

//merge

//Oscar Alejandro Penilla 20300701

#include <iostream>

using namespace std;

class numeros {

public:

int cantidad;

int arreglo[50];

//constructores y destructures y funciones

numeros(int, int\*);

~numeros();

void merge(int\*, int, int,int);

void merg\_recursi(int\*,int, int);

};

numeros::numeros(int, int\*)

{

numeros::cantidad = 0;

numeros::arreglo[50];

}

numeros::~numeros() { }

void numeros::merge(int\* mer\_arreglo, int mer\_ini, int mer\_mid, int mer\_fin) {

int i\_izq,j\_der,k\_arrfin;//contadores externos a los for

int elementosIzq = mer\_mid - mer\_ini + 1;

int elementosDer = mer\_fin - mer\_mid;

int izquierda[elementosIzq];//creamos arreglos para cada mitad

int derecha[elementosDer];

for(int i = 0; i < elementosIzq; i++){//lo recorremos y asignamos a nuestro nuevo arreglo

izquierda[i] = mer\_arreglo[mer\_ini+i];

}

for(int j = 0; j < elementosDer; j++){//mismo procedimiento

derecha[j] = mer\_arreglo[mer\_mid + 1 + j];

}

i\_izq = 0;//contador para inzquierda

j\_der = 0;//contador para la derecha

k\_arrfin = mer\_ini;//k va a ser igual al inicio

while(i\_izq < elementosIzq && j\_der < elementosDer){

if(izquierda[i\_izq] <= derecha[j\_der]){

mer\_arreglo[k\_arrfin] = izquierda[i\_izq];

i\_izq++;//aqui los usamos para nuestro while

}else{

mer\_arreglo[k\_arrfin] = derecha[j\_der];

j\_der++;

}

k\_arrfin++;//aumentamos contador de nuestro arreglo final

}

while(j\_der < elementosDer){

mer\_arreglo[k\_arrfin] = derecha[j\_der];

j\_der++;

k\_arrfin++;

}

while(i\_izq < elementosIzq){

mer\_arreglo[k\_arrfin] = izquierda[i\_izq];

i\_izq++;

k\_arrfin++;

}

}

void numeros::merg\_recursi(int\* arreglo,int ini, int fin){//merg\_recursi

if(ini < fin){

int mid = ini + (fin - ini)/2;

merg\_recursi(arreglo,ini,mid);

merg\_recursi(arreglo,mid+1,fin);

merge(arreglo,ini,mid,fin);

}

}

int main()

{

int cant = 0;

int arreglo[50];

do

{

cout << "Cual es la cantidad de digitos que quieres ingresar? Min 1 max 50" << endl;

cin >> cant;

if (cant < 1 || cant > 50) {

cout << "Valor fuera de rango\n";

}

} while (cant < 1 || cant > 50);

cout << "Ingresa los datos: " << endl;

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

{

cin >> arreglo[i];

}

numeros ob1(cant, arreglo);

ob1.merg\_recursi(arreglo,0, cant-1);

cout<<"Arreglo ordenado: ";

for(int i = 0; i < cant; i++){

cout << arreglo[i] << " ";

}

cout << endl;

}