Logotipo

Descripción generada automáticamente

# Método de merge

# 

**Fecha:** 14/03/2023

**Materia:** Estructura de datos

**Profesor/Profesora:** Ismael López Buenrostro

**Nombre:** Luis Rigoberto Ramírez Llamas

**Registro:** 21100298

**Carrera:** Drllo. Software

**Grupo:** 4E1

**Descripción del programa**

Este programa es un código en C++ a través de visual studio, en donde se introduce un conjunto 1 a 50 de números enteros ya sean ordenados o desordenados, positivos o negativos con la finalidad de ordenarlos con el método de merge

En este código se usa una clase Métodos y una función Merge a través de métodos y funciones recursivas para dividir el arreglo

**Lista de requerimientos**

Fabricar un programa que realice el proceso de ordenación por el método de Merge el cual deberá tener las siguientes características.

a) Ordenar 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 cada posibilidad.

g) Evitar 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?

A través del método de recursividad dividiendo el arreglo hasta tener solo 2 números el arreglo divido regresando y terminando el método de organización

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

Comparando dependientemente con los valores iniciales y finales, de subarreglos ordenándolos

**Contenido del PSP**

|  |  |  |  |
| --- | --- | --- | --- |
| Código fuente: | Estimado  (min) | Real  (min) | Error |
| Tiempo de Diseño | 50 | 60 | 20% |
| Tiempo de Codificación. | 180 | 200 | 11% |
| Tiempo estimado de fabricación | 180 | 230 | 27% |
| Total, de líneas de código nuevas. | 60 | 58 | 3% |
| Total, de líneas de código reusadas. | 30 | 28 | 6% |
| Total, de líneas de código del programa. | 90 | 86 | 4% |
| Cantidad de errores de compilación. | 20 | 15 | 25% |
| Cantidad de errores de ejecución. | 5 | 4 | 20% |

**Código Fuente**

#include <iostream>

using namespace std;

class Metodos

{

public:

void Merge(double Numeros[50], int Inicio, int Final);

};

void Metodos::Merge(double Numeros[50], int Inicio, int Final)

{

int Mitad = Inicio + (Final - Inicio) / 2;

if (Inicio < Final)

{

int PosIzq;

int PosDer;

Merge(Numeros, Inicio, Mitad);

Merge(Numeros, Mitad + 1, Final);

int Izq = Mitad - Inicio + 1;

int Der = Final - Mitad;

double\* NumIzq = new double[Izq];

double\* NumDer = new double[Der];

for (PosIzq = 0; PosIzq < Izq; PosIzq++)

{

NumIzq[PosIzq] = Numeros[Inicio + PosIzq];

}

for (PosDer = 0; PosDer < Der; PosDer++)

{

NumDer[PosDer] = Numeros[Mitad + 1 + PosDer];

}

PosIzq = 0;

PosDer = 0;

while (PosIzq < Izq && PosDer < Der)

{

if (NumIzq[PosIzq] <= NumDer[PosDer])

{

Numeros[Inicio + PosIzq + PosDer] = NumIzq[PosIzq];

PosIzq++;

}

else

{

Numeros[Inicio + PosIzq + PosDer] = NumDer[PosDer];

PosDer++;

}

}

while (PosDer < Der)

{

Numeros[Inicio + PosIzq + PosDer] = NumDer[PosDer];

PosDer++;

}

while (PosIzq < Izq)

{

Numeros[Inicio + PosIzq + PosDer] = NumIzq[PosIzq];

PosIzq++;

}

}

}

int main()

{

Metodos Merge;

double Numeros[50];

int Tam;

cout << "Cuantos numeros vas a ingresar: " << endl;

cin >> Tam;

if (Tam <= 0 || Tam > 50)

{

cout << "Tamaño no compatible" << endl;

return 0;

}

else

{

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

{

cout << "Introduce el " << i + 1 << " numero: " << endl;

cin >> Numeros[i];

}

}

cout << "Estos son tus numeros ordenados : \n" << endl;

Merge.Merge(Numeros, 0, Tam - 1);

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

{

cout << Numeros[i] << "\t";

}

cout << endl;

return 0;

}