

UNIVERSIDAD NACIONAL DE INGENIERÍA



Introducción a la Programación

Método de ordenamiento: Mergesort

Nombre: **HECTOR ANTONIO CANO OLIVA**

N.º Carnet: 2024-1655U

Docente: Ing. Cristopher Larios

Grupo: 1M7-S

Fecha: 05/05/2024

ciudad, Managua

Explicación de cómo se puede ocupar este método de ordenamiento de mergesort

En primer lugar, se debe ordenar la lista alfabéticamente de acuerdo al nombre de cada persona.

No obstante, en caso de que existan dos personas con el mismo nombre, entonces se deberá ordenar a esas personas de manera ascendente según su edad.

Inicialmente, se compararán los nombres de las personas que conforman la lista.

Cuando se presenten coincidencias en los nombres, se procederá a comparar las edades correspondientes.

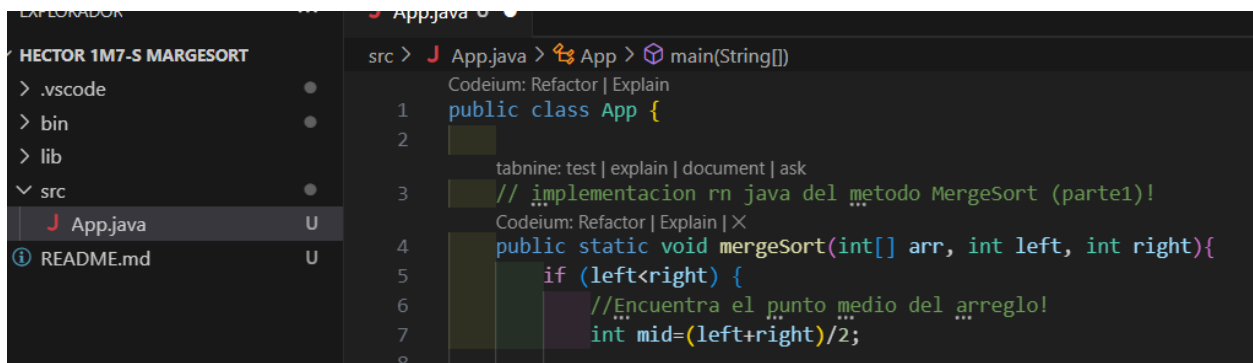
Posteriormente, se dividirá la lista en mitades y se ordenará cada una de ellas de forma recursiva.

Finalmente, se combinarán las mitades ordenadas, asegurándose de que los elementos queden correctamente ordenados.

Elaboración de la guía

Implementación en java “método mergesort” –(parte 1)

🔴 Primeramente, MergeSort divide el arreglo en mitades recursivamente.



```
src > J App.java > App > main(String[])
Codeium: Refactor | Explain
1 public class App {
2
3 // implementacion rn java del metodo MergeSort (parte1)!
Codeium: Refactor | Explain | X
4 public static void mergeSort(int[] arr, int left, int right){
5     if (left<right) {
6         //Encuentra el punto medio del arreglo!
7         int mid=(left+right)/2;
```

👹 Segundo llamadas recursivas para ordenar cada mitad.

```
HECTOR 1M7-S MARGESORT  src > J App.java > App > main(String[])
> .vscode
> bin
> lib
src
  J App.java U
  README.md U

1 public class App {
2
3     tabnine: test | explain | document | ask
4     // implementación rn java del método MergeSort (parte1)!
5     public static void mergeSort(int[] arr, int left, int right){
6         if (left<right) {
7             //Encuentra el punto medio del arreglo!
8             int mid=(left+right)/2;
9
10            //Ordena recursivamente la mitad izquierda!
11            void App.mergeSort(int[] arr, int left, int right)
12            mergeSort(arr, left, mid);
13            mergeSort(arr, mid + 1, right);
14
15            // Combina las dos mitades ordenadas!
16            merge(arr, left, mid, right);
17        }
18    }
```

Implementación en Java (Parte 2) - Método main

👹 El método Merge combina dos subarreglos ordenados (L y R) en un solo arreglo (arr).

```
HECTOR 1M7-S MA...  src > J App.java > App
> .vscode
> bin
> lib
src
  J App.java U
  README.md U

1 public class App {
2
3     tabnine: test | explain | document | ask | Codeium: Refactor | Explain | Generate Javadoc | X
4     public static void mergeSort(int[] arr, int left, int right){
5
6         // Combina las dos mitades ordenadas!
7         merge(arr, left, mid, right);
8     }
9
10    private static void merge(int[] arr, int left, int mid, int right){
11        //Tamaño de los subarreglos a fusionar!
12        int sizeLeft=mid-left+1;
13        int sizeRight=right-mid;
14
15        //Arreglos temporales para almacenar los subarreglos!
16        int[] tempLeft=new int[sizeLeft];
17        int[] tempRight=new int[sizeRight];
18
19        //Copia datos a los arreglos temporales!
20        for (int i = 0; i < sizeLeft; i++) {
21            tempLeft[i]=arr[left+i];
22        }
23        for (int j = 0; j < sizeRight; j++) {
24            tempRight[j]=arr[mid+1+j];
25        }
26
27        //Fusiona los subarreglos temporales en el arreglo original!
28        int i=0, j=0;
29        int k=left; //Índice inicial para el arreglo fusionado!
30
31        while (i<sizeLeft && j<sizeRight) {
32            if(tempLeft[i]<=tempRight[j]){
33                arr[k]=tempLeft[i];
34                i++;
35            } else{
36                arr[k]=tempRight[j];
37                j++;
38            }
39            k++;
40        }
41
42        //Copia elementos restantes de tempLeft[] si los hay!
43        while (i<sizeLeft) {
44            arr[k]=tempLeft[i];
45            i++;
46            k++;
47        }
48
49        //Copia elementos restantes de tempRight[] si los hay!
50        while (j<sizeRight) {
51            arr[k]=tempRight[j];
52            j++;
53            k++;
54        }
55    }
56 }
```

Implementación en Java (Parte 3) - Método main

Se crea un arreglo desordenado.

```
src > J App.java > App
1  public class App {
75  public static void main(String[] args) throws Exception {
76
77      int[] arr={38, 27, 43, 3, 9, 82, 10};
78
79      int n=arr.length;
80
81      mergeSort(arr, left:0, n-1); //Llamada al metodo de ordenamiento MergeSort!
82
83
84      System.out.println(x:"Arreglo ordenado: ");
85
86      for (int num: arr){
87          System.out.print(num + " ");
88      }
89
90
91
92 }
```

Llamamos al método MergeSort para ordenar el arreglo.

Imprimimos el arreglo ordenado.

Ejecución del Programa

```
SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  GITLENS  COMENTARIOS
esort\bin" App "
Arreglo ordenado:
3 9 10 27 38 43 82
C:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort>

C:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort>
C:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort> c: && cd "c:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort" && cmd /C ""C:\Program Files\Eclipse Adoptium\jdk-21.0.2.13-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort\bin" App "
Arreglo ordenado:
3 9 10 27 38 43 82
C:\Users\hcano\Desktop\guia de ordenamiento margsort\hector 1m7-s margsort>
```

Creación y Subida al Repositorio en GitHub

🔴 Maquina local para guardar en el repositorio.

```
hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (main)
$ git init
Initialized empty Git repository in C:/Users/hcano/Desktop/guia de ordenamiento
margsort/hector 1m7-s margsort/.git/

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git config --global user.email "hector.cano181@std.uni.edu.ni"

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git config --global user.name "héctor20061987"

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git remote add origin https://github.com/hector20061987/Guia-de-ordenamiento-m
argsort.git

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git add .


hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git commit
[master (root-commit) 40b8078] first commit 2
4 files changed, 117 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 bin/App.class
create mode 100644 src/App.java

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (master)
$ git branch -M main


hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 2.97 KiB | 1.49 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hector20061987/Guia-de-ordenamiento-margsort.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

hcano@DESKTOP-J5FI0FL MINGW64 ~/Desktop/guia de ordenamiento margsort/hector 1m
7-s margsort (main)
$
```


🔴 Subida del proyecto en GitHub.

 hector20061987 / Guia-de-ordenamiento-mergesort

[Código](#) [Asuntos](#) [Solicitudes de extracción](#) [Comportamiento](#) [Proyectos](#) [wiki](#) [Seguridad](#) [Perspectivas](#) [Ajustes](#)

 **Guia-de-ordenamiento-mergesort** Público Alfiler Dejar de mirar 1

principal 1 sucursal 0 etiquetas Agregar archivo Código


 hector20061987

primer compromiso 2


40b8078 · hace 3 minutos

🕒 1 se compromete

📁 .vscode	primer compromiso 2	hace 3 minutos
📁 papelera	primer compromiso 2	hace 3 minutos
📁 src	primer compromiso 2	hace 3 minutos
📄 LÉAME.md	primer compromiso 2	hace 3 minutos

 hector20061987

Principales repositorios Nuevo

 hector20061987 / Guia-de-ordenamiento-de-insercionSort

Actividad reciente

Cuando realiza acciones en GitHub, le proporcionaremos enlaces a esa actividad aquí.

brillante diseño visual. ✨

[Aprende más](#)

<> Empezar a escribir código

Iniciar un nuevo repositorio para hector20061987

Un repositorio contiene todos los archivos de su proyecto, el historial de revisiones y las discusiones de los colaboradores.

Nombre del repositorio *

✔️ Your new repository will be created as Guia-de-ordenamiento-mergesort.

The repository name can only contain ASCII letters, digits, and the characters ., - and _.

☒ **Público**

Cualquier persona en Internet puede ver este repositorio.

☐ **Privado**

Tú eliges quién puede ver y comprometerse con este repositorio.

Crear un nuevo repositorio

GITHUB