

**Documentación Práctica 7**  
**Estructura de Datos Lineales**  
**Recursividad.**  
**Búsqueda Binaria Recursiva y Quicksort Recursivo.**

**Introducción.**

La idea básica detrás de un algoritmo recursivo es que para resolver el problema hay que dividir el problema en sub-problemas, los cuales son instancias más pequeñas del problema, y usar la solución del sub-problema para resolver el problema original.

Para que un algoritmo recursivo funcione, el sub-problema debe llegar en algún punto a un caso base. Así que cada vez que se haga uso de la recursividad esta debe de realizarse en una instancia más pequeña del mismo problema y llegar a un caso base que al final se resuelve sin el uso de la recursividad.

**Descripción del Problema.**

Con el uso de la recursividad de realizarán los métodos de búsqueda binaria y el algoritmo de ordenamiento Quicksort.

Se procederá a crear una fila de N números y se ordenaran por el método Quicksort y posteriormente se buscará una llave con el uso del método búsqueda binaria. Al finalizar dichas tareas se mostrará en consola la lista ordenada, sin es que el usuario lo desea, y si la llave fue encontrada o no en la fila.

**Desarrollo.**

A continuación, se mostrarán los pseudocódigos de los métodos que se implementarán para resolver el problema.

En está practica los datos que puede recibir un nodo pueden ser diferentes, la estructura del nodo es prácticamente la misma solo se le agregó un nuevo tipo de variable, esta variable es la llave del nodo. La llave del nodo es de tipo entero mientras que el dato del nodo puede ser de diferente tipo, ya que puede ser caractere, entero, flotante o boléano.

Los pseudocódigos de la creación de un nodo e insertar nodo son los mismos que se han usado en prácticas anteriores.

### **Pseudocódigo Búsqueda Binaria Recursivo.**

- Entrada: I, F  
Salida: Fila.

Función nodo BúsquedaBinaria(I, F, llave)

Inicio

NODO Mitad, Inicio, Fin, Aux

Inicio <- I

Fin <- F

Si (Aux <> Inicio) entonces

Si (Inicio = 0) entonces

Retorna 0

Fin Si

Mitad <- Inicio

Fin <- Inicio.Sig

Mientras (Fin <> Aux) Hacer

Fin <- Fin.Sig

Si (Fin <> Aux) entonces

Mitad <- Mitad.Sig

Fin <- Fin.Sig

Fin Si

Fin Mientras

Si (Mitad = 0) entonces

Retorna 0

Fin Si

Si (Mitad.Llave = llave) entonces

Retorna Mitad

Otro Si (Mitad.Llave < llave) entonces

Retorna Llamada a Función BúsquedaBinaria(Mitad.Sig, F, llave)

Otro

Retorna Llamada a Función BúsquedaBinaria(I, Mitad, llave)

Fin Si

Fin Si

Retorna 0

Fin

Fin Función

### **Pseudocódigo Quicksort Recursivo.**

- Entrada: I, F.  
Salida: Fila.

Función Vacía Quicksort(I, F)

Inicio

NODO a, A, P, Aux

Booleano bandera

a <- I

P <- I

Si (I = F)

Retorna

Fin Si

Mientras (a.Sig <> F) Hacer

Si ((a.Sig).Llave < P.Llave) entonces

A <- I

I <- a.Sig

A.Ant <- I

A.Sig <- (a.Sig).Sig

I.Sig <- A

I.Ant <- 0

Otro

a <- a.Sig

Fin Si

Fin Mientras

Llamada a Función Quicksort(I, P)

Llamada a Función Quicksort(P.Sig, F)

Fin

Fin Función

## Resultados.

Se crea un arreglo de N números ordenados y sin repetir y con el uso de dos arreglos auxiliares se crea un cuarto arreglo que contiene los números ahora de una manera desordenada, cada índice del arreglo posee un número que posteriormente se usara como la llave y el dato de cada nodo que se creará.

Con un ciclo “para” se crea la fila de N nodos con su respectiva llave y dato, posteriormente con el método Quicksort se ordenará la fila, para posteriormente buscar una llave ya predeterminada y mostrar en consola si esta se encuentra o no en la fila.

## Ejemplo.

**Llave: 19**

### ***Lista Desordenada:***

15 12 39 2 7 9 35 3 10 5 1 70 11 8 17 4

### ***Lista Ordenada:***

1 2 3 4 5 7 8 9 10 11 12 15 17 35 39 70

**La llave 19 no se encuentra en la fila.**

## Manual de Usuario.

- Para compilar este programa es necesario escribir esto en la terminal de LINUX:
  - `cd /UBICACIÓN ARCHIVO/`
  - Ya ubicado en la carpeta dónde se encuentre el main.cpp se escribe:
    - `make all`
  - Por ejemplo:
    - `cd /home/NombreUsuario/CarpetaArchivo/`
- Para ejecutar el programa se escribe lo siguiente en la terminal de LINUX:
  - `./main`