



UNIVERSIDAD TECNOLÓGICA DE  
PANAMÁ CENTRO REGIONAL DE  
CHIRIQUÍ  
FACULTAD DE INGENIERÍA DE SISTEMAS  
COMPUTACIONALES



**CARRERA:**  
Gestión y Desarrollo de Software

ACTIVIDAD No. 19

PARCIAL No. 3

“Parcial #3”

**ASIGNATURA:** Estructura de Datos II

**DOCENTE:**  
Profa. Nunehar Mondul

**ESTUDIANTE/s:**  
Jorge Jiménez (4-826-874)

*I SEMESTRE 2025*

**FECHA:**  
07/08/2025

## Desarrollo

1. "Ordenar la siguiente lista con el método de **inserción** o burbuja y árbol binario (20 puntos)."

3	5	20	4	8	13
---	---	----	---	---	----

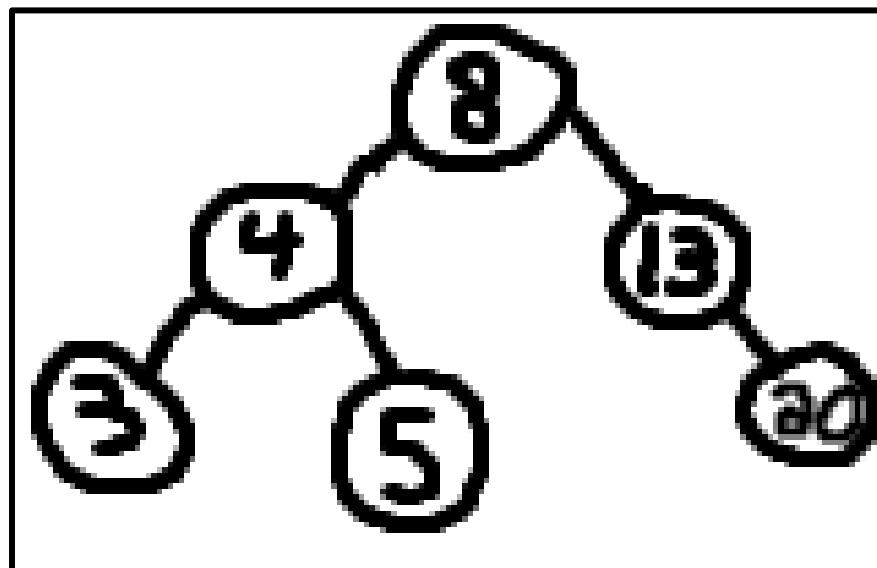
Inserción:

3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	5	20	4	8	13
3	4	5	20	8	13
3	4	5	20	8	13
3	4	5	20	8	13
3	4	5	20	8	13
3	4	5	20	8	13

3	4	5	20	8	13
3	4	5	8	20	13
3	4	5	8	20	13
3	4	5	8	13	20

---

Árbol Binario:

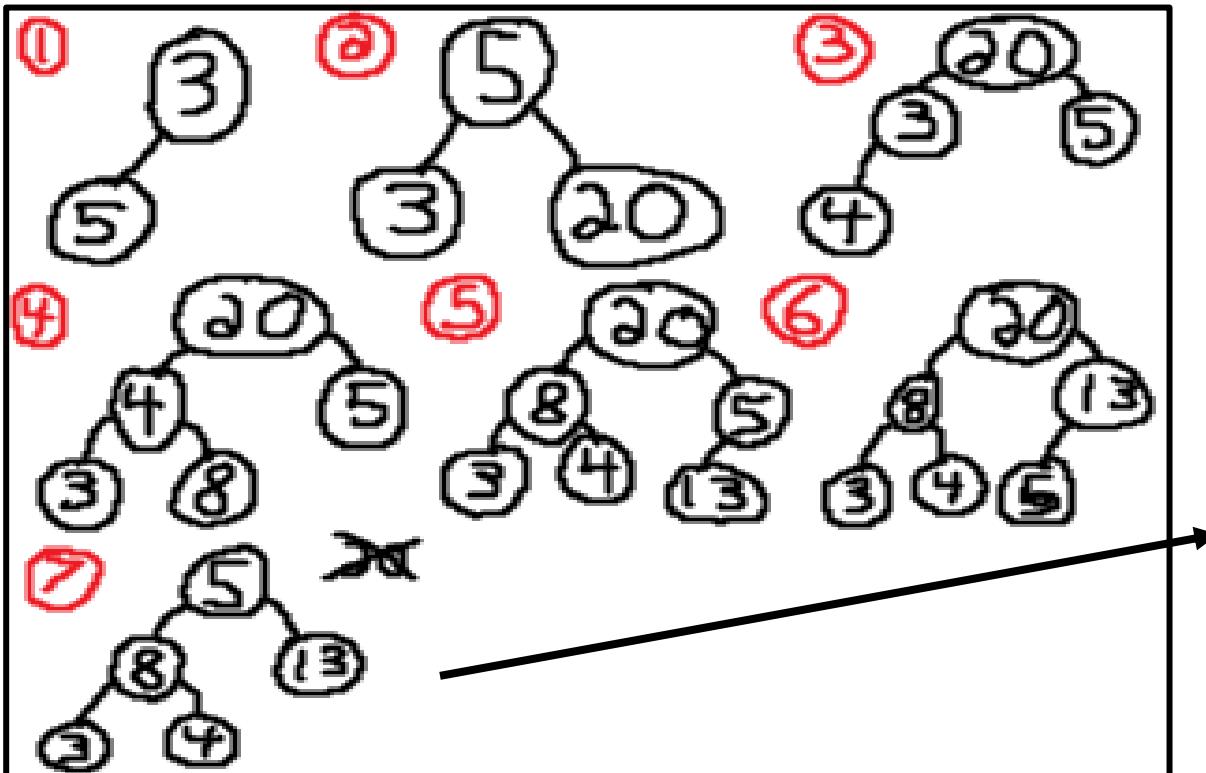


Utilizado el método (IRD):  
Izquierda, Raíz, Derecha.

2. "Ordenar la lista de la 1 con Quicksort (**15 puntos**)."

3	5	20	4	8	13
Pivot: 3					
3	5	20	4	8	13
Pivot: 5					
3	4	5	20	8	13
Pivot: 20					
3	4	5	8	13	20
Pivot: 8					
3	4	5	8	13	20

3. "Ordenar la lista de la 1 con Heapsort (**15 puntos**)."



Los pasos en orden:

- 1) [3, 5, 20, 4, 8, 13]
- 2) [5, 3, 20, 4, 8, 13]
- 3) [20, 3, 5, 4, 8, 13]
- 4) [20, 4, 5, 3, 8, 13]
- 5) [20, 8, 5, 3, 4, 13]
- 6) [20, 8, 13, 3, 4, 5]
- 7) [5, 8, 13, 3, 4, 20]

Y así continuaría (*por el tiempo y el espacio, deje de dibujar*):

- 8) [13, 8, 5, 3, 4, 20]
- 9) [4, 8, 5, 3, 13, 20]
- 10) [8, 4, 5, 3, 13, 20]
- 11) [3, 4, 5, 8, 13, 20]

4. "Hacer tabla donde mencione la notación Big O de cada uno de los métodos utilizados y cuál es su diferencia en términos de tiempo ([15 Puntos](#))."

Método	Mejor Big O	Diferencia en tiempo
Inserción	$O(n)$	Muy eficiente con listas pequeñas o casi ordenadas, pero lento con listas grandes.
Quicksort	$O(n \log n)$	Muy eficiente en promedio, pero tiene el peor caso si se elige mal el pivote.
Heapsort	$O(n \log n)$	También es eficiente, pero más lento que Quicksort por constantes mayores.

5. "Mencione uno de los algoritmos de búsqueda en cadena o párrafo y cómo funciona ([10 puntos](#))."

R: "Knuth-Morris-Pratt" porque busca una palabra en un texto sin revisar letras dos veces. Usa una tabla para saber desde dónde continuar si hay un error, lo que hace la búsqueda más rápida y eficiente.

6. "Aplicar búsqueda binaria de número 7 a la siguiente lista. Mencionar que debemos realizar previamente y los comandos en java ([15 puntos](#))."

6	5	3	1	8	7	2	4
---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Se ordena primero.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

4 es menor que 7 así que va por la derecha. Los otros menores no se cuentan.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

6 es menor que 7 así que va por la derecha de nuevo.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

El 7 se encuentra en el nuevo centro en medio del 6 y 8.

Antes, si me acuerdo, se debe de ordenar la lista de menor a mayor, ya que este tipo de búsqueda funciona mejor y solamente en listas ordenadas.

#### **Comandos en Java:**

Arrays.sort(array); para ordenar la lista.

Arrays.binarySearch(array, valor); para buscar el número.

7. “El siguiente algoritmo, ¿Qué tipo de ordenamiento realiza? sustente **(10 puntos)**.”

El Código realiza "**Quicksort**". Se está comparando cada elemento con el pivote. Si el valor es menor o igual, lo intercambia hacia la izquierda del arreglo, manteniendo separados los menores y mayores al pivote.

```
for (int j = low; j < high; j++) {  
    if (array[j] <= num) { i++;  
  
        int temp = array[i];  
        array[i] = array[j];  
  
        array[j] = temp;  
    }  
}
```