UNIDAD TEMÁTICA 9 – CLASIFICACIÓN Trabajo de Aplicación 3

1. Selección del primer elemento como pivote

La primera y más sencilla forma de seleccionar el pivote es tomar el primer elemento del arreglo. Este método es fácil de implementar y no requiere cálculos adicionales. Sin embargo, su eficiencia depende del orden inicial de los elementos en el arreglo.

Análisis del tiempo de ejecución:
En el peor de los casos, si el arreglo ya está ordenado o casi ordenado, el tiempo de ejecución de Quicksort con este método es $O(n^2)$. Esto se debe a que siempre se divide el arreglo de manera desequilibrada, lo que resulta en muchas recursiones profundas.

Conveniencia comparativa:
Este método no es eficiente para arreglos grandes o cuando se sospecha que los datos están ordenados o inversamente ordenados. Sin embargo, es una buena opción para arreglos pequeños o cuando la simplicidad del código es prioritaria

Seudocódigo(Primer elemento):

```
Método encuentraPivote(arr, bajo, alto):
    retorno bajo
```

2. Selección del pivote aleatorio

La segunda forma es seleccionar el pivote de manera aleatoria dentro del rango de índices del arreglo. Esta técnica ayuda a evitar el peor de los casos de $O(n^2)$ al garantizar que la división del arreglo sea más balanceada en promedio.

Análisis del tiempo de ejecución:
El tiempo de ejecución esperado para este método es $O(n \log n)$. Aunque en el peor de los casos todavía es $O(n^2)$, la probabilidad de que esto ocurra es baja debido a la aleatoriedad en la selección del pivote.

Conveniencia comparativa:
Este método es generalmente más eficiente que la selección del primer elemento, especialmente para arreglos grandes y desordenados. Sin embargo, la implementación es un poco más compleja debido a la necesidad de generar números aleatorios.

Seudocódigo(Elemento Aleatorio):

```
Método encuentraPivote(arr, bajo, alto):
    índiceAleatorio = aleatorio(bajo, alto)
    intercambiar(arr, bajo, índiceAleatorio)
retorno bajo
```

3. Selección de la mediana de tres

La tercera forma es seleccionar la mediana de tres elementos: el primero, el medio y el último del arreglo. Este método ayuda a mitigar los efectos negativos de un arreglo ya ordenado y generalmente proporciona una buena partición del arreglo.

Análisis del tiempo de ejecución:
El tiempo de ejecución esperado para este método es O(n log n), similar al método del pivote aleatorio. En el peor de los casos, el tiempo de ejecución sigue siendo $O(n^2)$, pero es menos probable que ocurra debido a la selección más robusta del pivote.

Conveniencia comparativa:
Este método es eficiente y relativamente fácil de implementar. Es adecuado para la mayoría de los casos, ya que combina la simplicidad con una mejor garantía de rendimiento.

Seudocódigo(Selección Mediana):

```
Método encuentraPivote(arr, bajo, alto):
    medio = (bajo + alto) / 2
    si arr[bajo] > arr[medio]:
        intercambiar(arr, bajo, medio)
    si arr[bajo] > arr[alto]:
        intercambiar(arr, bajo, alto)
    si arr[medio] > arr[alto]:
        intercambiar(arr, medio, alto)
    intercambiar(arr, medio, bajo)
retorno bajo
```