UT9 – TA3

Forma 1: primer elemento de pivote

Análisis del tiempo de ejecución

* **Peor caso:** O(n²), ocurre cuando el arreglo ya está ordenado o casi ordenado.
* **Mejor caso:** O(n logn), ocurre cuando los subarreglos son aproximadamente del mismo tamaño en cada partición.
* **Caso promedio:** O(n logn), generalmente es más eficiente en la mayoría de los casos aleatorios

Conveniencia comparativa

* **Ventajas:** Es simple de implementar y tiene bajo costo computacional.
* **Desventajas:** Puede llevar a un rendimiento deficiente en casos de datos ya ordenados.

funcion elegirPivote(array, inicio, fin):

return array[inicio]

Forma 2: pivote al azar

Análisis del tiempo de ejecución

* **Peor caso:** O(n²), aunque menos probable que ocurra.
* **Mejor caso**: O(n log n).
* **Caso promedio:** O(n log n), reduce la probabilidad de caer en el peor caso.

Conveniencia comparativa

* **Ventajas:** Reduce la probabilidad de particiones desbalanceadas, mejor rendimiento en promedio.
* **Desventajas:** La elección del pivote aleatorio implica una llamada a la función de generación de números aleatorios, lo que puede añadir un pequeño costo computacional adicional.

función elegirPivoteAzar(array, inicio, fin):

elementoRandom = random.randint(inicio, fin)

swap(array, inicio, randomIndex)

return array[low]

3. Mediana de Tres

Análisis del tiempo de ejecución

* **Peor caso:** O(n²), aunque menos probable.
* **Mejor caso:** O(n log n).
* **Caso promedio:** O(n log n), generalmente proporciona particiones más balanceadas.

Conveniencia comparativa

* **Ventajas:** Mayor probabilidad de obtener particiones equilibradas, mejor rendimiento en la práctica.
* **Desventajas:** Mayor costo computacional en comparación con los métodos anteriores debido al cálculo de la mediana.

función elegirPivoteMediana(array, inicio, fin):

med = (inicio + fin)

if array[inicio] > array[med]:

swap(array, inicio, med)

if array[inicio] > array[fin]:

swap(array, inicio, fin)

if array[med] > array[fin]:

swap(array, med, fin)

swap(array, med, fin - 1)

return array[fin - 1]