Ut9- PD3

1)

 **División y Pivote**: El algoritmo selecciona un elemento pivote de la lista de elementos a ordenar.

* **Condición Invariante**: En cada llamada recursiva de quicksort, los elementos del subarreglo están divididos de manera que todos los elementos a la izquierda del pivote son menores que el pivote, y todos los elementos a la derecha son mayores o iguales al pivote.

 **Partición**: El arreglo se reordena de forma tal que los elementos menores que el pivote queden antes del pivote y los elementos mayores o iguales queden después. Esto se logra con un índice k que marca el punto de partición.

* **Condición Invariante**: Después de la partición, para cualquier índice k tal que i≤k<j, se cumple que los elementos V[i] a V[k−1] son menores que el pivote y los elementos V[k] a V[j] son mayores o iguales al pivote.

 **Recursión**: Quicksort se llama recursivamente para ordenar las dos sublistas que se generan después de la partición.

* **Condición Invariante**: La recursión asegura que cada subarreglo se ordena correctamente al aplicar la misma lógica de pivote y partición, manteniendo la condición invariante en cada nivel de recursión.

2)

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

3)

**Particionamiento en tres partes**:

Este enfoque divide el array en tres partes: elementos menores, iguales y mayores que un pivote. Esto es especialmente eficiente para conjuntos de datos con muchos elementos duplicados.

**Orden del tiempo de ejecución**:

El tiempo de ejecución promedio del QuickSort de tres vías es O(nlog^n), similar al QuickSort tradicional. Sin embargo, en el peor de los casos, el tiempo de ejecución puede ser O(n^2). La ventaja principal se observa cuando tenemos conjuntos de datos con muchos elementos repetidos, donde el rendimiento mejora significativamente comparado con el QuickSort tradicional

**Conjuntos de datos ordenados de diferentes formas**:

* **Datos aleatorios**: El tiempo de ejecución es O(nlog^n).
* **Datos casi ordenados o inversamente ordenados**: Aunque el QuickSort tradicional puede degradarse a O(n2) el QuickSort de tres vías maneja mejor estos casos debido al particionamiento más granular.
* **Conjuntos con claves duplicadas**: Sobresale el QuickSort de tres vías, ya que el particionamiento en tres partes reduce significativamente el número de comparaciones y movimientos necesarios.

**4)**

**Peor caso:**

El peor caso para Quicksort ocurre cuando el pivote seleccionado no divide el arreglo de manera equilibrada. En el peor de los casos, esto sucede cuando el pivote es siempre el elemento más pequeño o más grande del subarreglo, lo que lleva a una partición desequilibrada. En este escenario, una de las particiones tendrá n−1 elementos y la otra partición estará vacía. Esto ocurre cuando el arreglo ya está ordenado o inversamente ordenado y siempre se selecciona el primer o último elemento como pivote.

El tiempo de ejecución va a ser: T(n)=O(n^2)

.