



INSTITUTO POLITÉCNICO NACIONAL
Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas.

REPORTE
METODO QUICKSORT

MATERIA:
Análisis y Diseño de Algoritmos

DOCENTE:
Erika Sanchez Femat

GRUPO:
3CM2

ALUMNA:
America Lizbet Flores Alcala

Introduccion

En el siguiente reporte se abordaran en tema del algoritmo QuickSort, del cual explicare, lo que es, como funciona, su complejidad y el mejor, peor y caso promedio.

Este método fue creado por el científico británico Charles Antony Richard Hoare, también conocido como Tony Hoare en 1960, su algoritmo Quicksort es el algoritmo de ordenamiento más ampliamente utilizado en el mundo.

¿QUE ES?

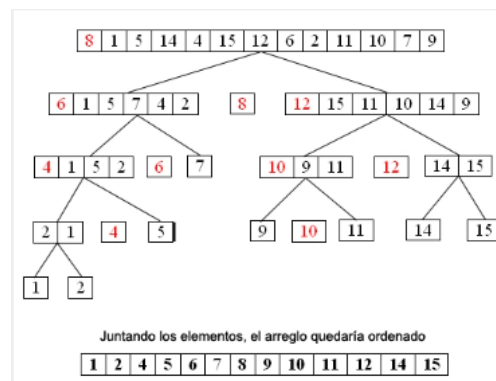
El método Quick Sort es actualmente el mas eficiente y veloz de los métodos de ordenación interna. Es también conocido con el nombre del método rápido y de ordenamiento por partición. Este método es una mejora sustancial del método de intercambio directo y recibe el nombre de Quick Sort, por la velocidad con la que ordena los elementos del arreglo. Quicksort es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar n elementos en un tiempo proporcional a $n \log n$.

Quicksort es actualmente el más eficiente y veloz de los métodos de ordenación interna.

¿COMO FUNCIONA?

Explicación abstracta del funcionamiento de QuickSort:

1. Se elige un elemento v de la lista L de elementos al que se le llama pivote.
2. Se particiona la lista L en tres listas:
 - L1 - que contiene todos los elementos de L menos v que sean menores o iguales que v
 - L2 - que contiene a v
 - L3 - que contiene todos los elementos de L menos v que sean mayores o iguales que v
3. Se aplica la recursión sobre L1 y L3
4. Se unen todas las soluciones que darán forma final a la lista L finalmente ordenada. Como L1 y L3 están ya ordenados, lo único que tenemos que hacer es concatenar L1, L2 y L3



Aunque este algoritmo parece sencillo, hay que implementar los pasos 1 y 3 de forma que se favorezca la velocidad de ejecución del algoritmo.

MEJOR, PEOR Y CASO PROMEDIO

$X=3$	a_1	a_2	a_3	a_4	a_5	a_6
$a_j = a_6 < X$	3	6	1	4	5	2
	$\uparrow i$					$\uparrow j$
	2	6	1	4	5	3
	$\uparrow i$					$\uparrow j$
$a_i = a_2 > X$	2	6	1	4	5	3
		$\uparrow i$				$\uparrow j$
	2	3	1	4	5	6
		$\uparrow i$				$\uparrow j$
	2	3	1	4	5	6
		$\uparrow i$			$\uparrow j$	
	2	3	1	4	5	6
		$\uparrow i$		$\uparrow j$		

	FORKING				FORKING	
$a_j = a_3 < X$	2	3	1	4	5	6
		$\uparrow i$	$\uparrow j$			
	2	1	3	4	5	6
		$\uparrow i$	$\uparrow j$			
	2	1	3	4	5	6
			$i \uparrow j$			
	$ \leftarrow \leq 3 \rightarrow $		$= 3$	$ \leftarrow \geq 3 \rightarrow $		

1. Mejor Caso:

El mejor caso de quick sort ocurre cuando X divide la lista justo en el centro. Es decir, X produce dos sublistas que contienen el mismo número de elementos. En este caso, la primera ronda requiere n pasos para dividir la lista original en dos listas. Para la ronda siguiente, para cada sublista, de nuevo se necesitan $n/2$ pasos (ignorando el elemento usado para la división). En consecuencia, para la segunda ronda nuevamente se requieren $2 * n/2 = n$ pasos. Si se supone que $n = 2^P$, entonces en total se requieren $p * n$ pasos. Sin embargo, $p = \log_2 n$. Así, para el mejor caso, la complejidad temporal del quick sort es $O(n \log n)$.

2. Peor caso:

El peor caso del quick sort ocurre cuando los datos de entrada están ya ordenados o inversamente ordenados. En estos casos, todo el tiempo simplemente se está seleccionando el extremo (ya sea el mayor o el menor). Por lo tanto, el número total de pasos que se requiere en el quick sort para el peor caso es:

$$n + (n - 1) + \dots + 1 = n/2(n + 1) = O(n^2).$$

- ### 3. Caso Promedio:

Para analizar el caso promedio, sea $T(n)$ que denota el número de pasos necesarios para llevar a cabo el quick sort en el caso promedio para n elementos. Se supondrá que después de la operación de división la lista se ha dividido en dos sublistas. La primera de ellas contiene s elementos y la segunda contiene $(n - s)$ elementos. El valor de s varía desde 1 hasta n y es necesario tomar en consideración todos los casos posibles a fin de obtener el desempeño del caso promedio. Para obtener $T(n)$ es posible aplicar la siguiente fórmula:

$$T(n) = Promedio(T(s) + T(n - s)) + cn \text{ con } l \leq s \leq n$$

donde cn denota el número de operaciones necesario para efectuar la primera operación de división. (Cada elemento es analizado antes de dividir en dos sublistas la lista original). Al resolver matemáticamente, se obtiene una eficiencia $O(n \log n)$.

Conclusion

En conclusión, Quick Sort es un algoritmo de ordenación eficiente basado en la estrategia de divide y vencerás, siendo el más rápido en promedio. Su desempeño varía según la disposición de los datos, con un mejor caso de $O(n \log n)$, un peor caso de $O(n^2)$ y un caso promedio de $O(n \log n)$. Esta versatilidad lo ha convertido en uno de los algoritmos de ordenamiento más ampliamente utilizados en el mundo para la clasificación de datos.