

Complejidad Temporal:

$$n = r - p$$

Línea	Instrucción	# veces que se ejecuta
	Partition (A,p,r)	
1	x = A[r]	1
2	i = p - 1	1
3	for j = p to r - 1	n+1
4	if A[j] ≤ x	n
5	i = i + 1	n
6	A[i] ↔ A[j]	n
7	A[i + 1] ↔ A[r]	1
8	return i + 1	1

Peor:

$$T(n) = 1 + 1 + (n + 1) + n + n + n + 1 + 1 = 5 + 4n = O(n)$$

$O(n)$

Mejor:

$$T(1) = \Omega(1) \rightarrow \text{el for no se ejecuta}$$

$\Omega(1)$

Promedio:

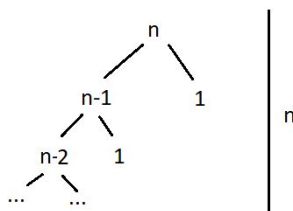
$\Theta(n)$

Línea	Instrucción	# veces que se ejecuta
	QuickSort (A,p,r)	
1	if p < r	1
2	q = Partition(A,p,r)	O(n)
3	QuickSort(A,p,q-1)	T(n/2)
4	QuickSort(A,q+1,r)	T(n/2)

Peor:

$$T(n) = 1 + O(n) + T(n-1) + T(1) \approx O(n) + T(n-1) + T(1)$$

$$n - i/2 = 1 \rightarrow i = 2(n-1) = 2n - 2 \approx n$$



$$\sum_{i=0}^n n = O(n^2)$$

$O(n^2)$

Mejor:

$\Omega(1) \rightarrow$ el if es falso

Promedio:

$$T(n) = 1 + O(n) + 2T(n/2) \approx O(n) + 2T(n/2)$$

por Método del maestro:

$$T(1) \rightarrow \Theta(1), a = 2, b = 2, a = b^1 \text{ es verdadero}$$

$$T(n) = \Theta(n^1 \log n) = \Theta(n \log n)$$

$\Theta(n \log n)$

Línea	Instrucción	# veces que se ejecuta
	Rand-Parti (A,p,r)	
1	i = Random(p,r)	1
2	A[r] \leftrightarrow A[i]	1
3	return Partition(A,p,r)	O(n)

Peor:

$$T(n) = 1 + 1 + O(n) = O(n)$$

$O(n)$

Mejor:

$$T(1) = \Omega(1) \rightarrow \text{se ejecuta Partition en el mejor caso}$$

$\Omega(1)$

Promedio:

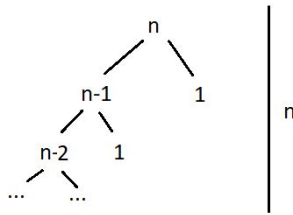
$\Theta(n)$

Línea	Instrucción	# veces que se ejecuta
	Randomized-QS (A,p,r)	
1	if p < r	1
2	q = Rand-Parti(A,p,r)	O(n)
3	Randomized-QS(A,p,q-1)	T(n/2)
4	Randomized-QS(A,q+1,r)	T(n/2)

Peor:

$$T(n) = 1 + O(n) + T(n-1) + T(1) \approx O(n) + T(n-1) + T(1)$$

$$n - i2 = 1 \rightarrow i = 2(n-1) = 2n - 2 \approx n$$



$$\sum_{i=0}^n n = O(n^2)$$

$O(n^2)$

Mejor:

$\Omega(1) \rightarrow$ el if es falso

Promedio:

$$T(n) = 1 + O(n) + 2T(n/2) \approx O(n) + 2T(n/2)$$

por Método del maestro:

$$T(1) \rightarrow \Theta(1), a = 2, b = 2, a = b^1 \text{ es verdadero}$$

$$T(n) = \Theta(n^1 \log n) = \Theta(n \log n)$$

$\Theta(n \log n)$

Unidad experimental:

-Medir el tiempo que tarda en ejecutarse los métodos de ordenamiento (QuickSort) junto a su auxiliar Partition, dado el aumento de objetos dentro de su arreglo(Array) y el espacio en memoria ram que ocupan estos objetos.

QuickSort? Array?

Variables de respuesta:

-El tiempo que se demora el programa en ejecutarse

Factores controlables:

-El espacio de memoria asignada en Visual Studio.

-El tamaño de los arrays.

-

Factores NO controlables:

-Los componentes del hardware del computador.

-La ejecución del programa dentro del computador.

-

Factores estudiados: