

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, Mejor y Promedio:

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

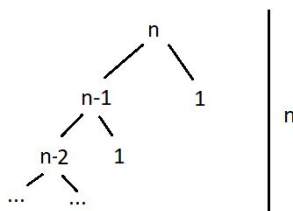
$O(n)$, $\Omega(1)$, $\Theta(n)$

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:

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

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



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

$$O(n^2)$$

Promedio y Mejor:

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

por Método del maestro:

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

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

$$\Theta(n \log n), \Omega(n \log n)$$

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

Peor, Promedio y Mejor:

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

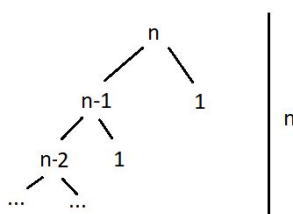
$$O(n), \Omega(n), \Theta(n)$$

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:

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

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



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

$O(n^2)$

Promedio y Mejor:

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

por Método del maestro:

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

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

$$\Theta(n \log n), \Omega(n \log n)$$

Unidad experimental:

-Algoritmo Quicksort

Variables de respuesta:

-El tiempo que se demora el método en ejecutarse

Factores controlables:

- **Estudiados:**

- Estado del arreglo.
- El tamaño de los arrays.
- Estado del algoritmo.

- **No Estudiados:**

- Cantidad de programas ejecutándose
- Capacidad de la memoria RAM

Factores NO controlables:

-La ejecución del programa dentro del computador.

Tratamientos Teóricos:

Prueba	Variante	Estado	Tamaño (n)	Tiempo (ms)
1	Normal	Ascendente	10	$O(n \log n)$
2			100	$O(n \log n)$
3			1000	$O(n \log n)$
4			10000	$O(n \log n)$
6		Descendente	10	$O(n^2)$

7		e	100	$O(n^2)$
8			1000	$O(n^2)$
9			10000	$O(n^2)$
11			10	$\Theta(n \log n)$
12		Random	100	$\Theta(n \log n)$
13			1000	$\Theta(n \log n)$
14			10000	$\Theta(n \log n)$
16			10	$\Theta(n \log n)$
17		Ascendente	100	$\Theta(n \log n)$
18			1000	$\Theta(n \log n)$
19			10000	$\Theta(n \log n)$
21			10	$\Theta(n \log n)$
22		Descendente	100	$\Theta(n \log n)$
23			1000	$\Theta(n \log n)$
24			10000	$\Theta(n \log n)$
26			10	$\Theta(n \log n)$
27			100	$\Theta(n \log n)$
28			1000	$\Theta(n \log n)$
29	Random	Random	10000	$\Theta(n \log n)$

1.b. Hasta ahora las etapas de estudio y diseño de experimentos que se han llevado a cabo hasta el momento son la planeación y realización. Las etapas que faltan son el análisis, la interpretación y el control y conclusiones finales.

1.c. El objetivo de este programa es comparar dos o más tratamientos, puesto que en este experimento hicimos varios tratamientos para ver cómo se altera la variable de respuesta cada vez y para analizar la varianza utilizando ANOVA. Esto con el fin de conocer el comportamiento del QuickSort y poder sacar una conclusión válida.

1.d.