

MÉTODO DE ORDENAMIENTO

QUICK SORT

ESTRUCTURA DE DATOS

5°I1

INTEGRANTES

NESTOR ARIEL GLEZ. HINOJOSA

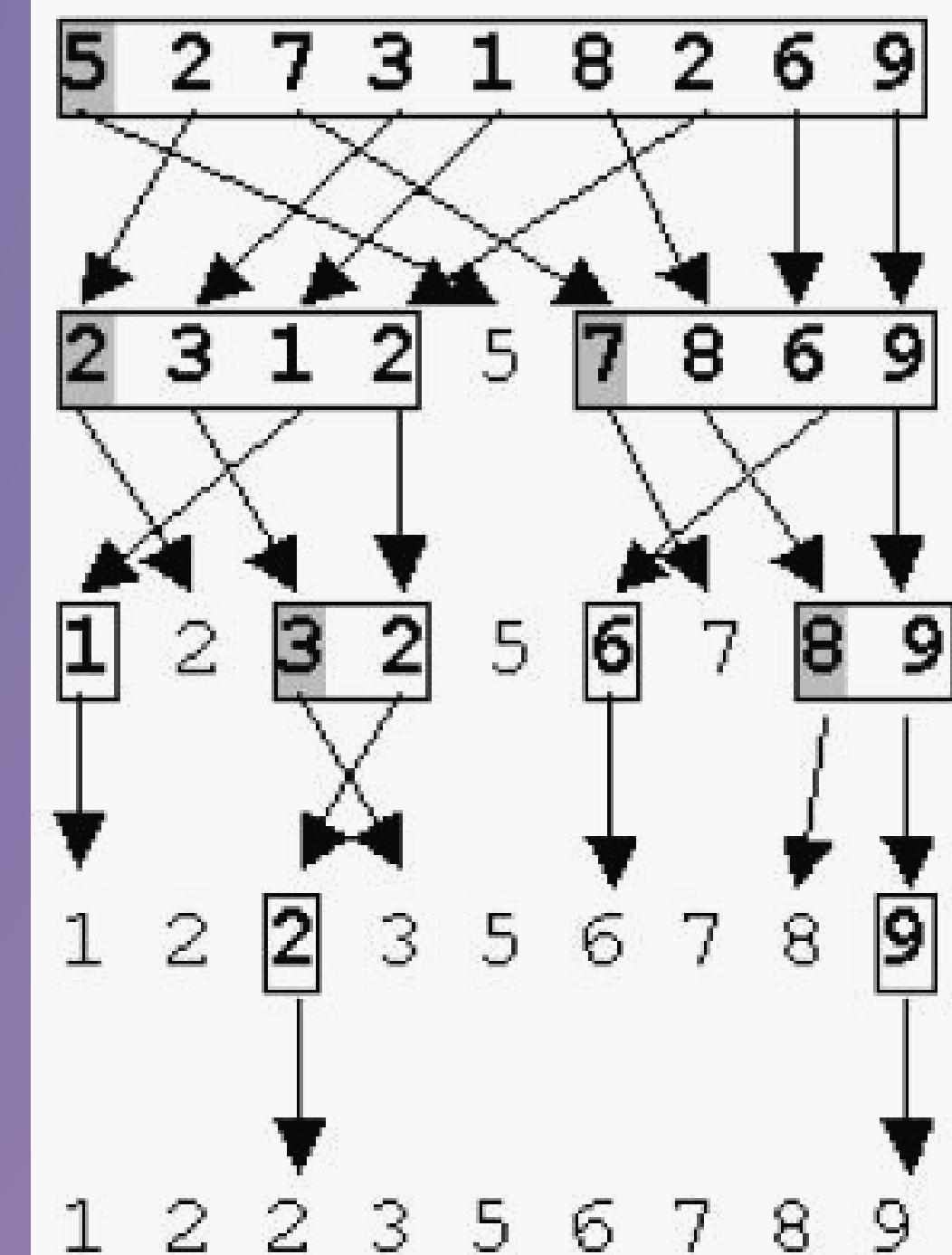
EMMANUEL BUENROSTRO BRISEÑO

JOSE GUILLERMO OLVERA PALACIOS

INTRODUCCIÓN

El método Quick Sort fue desarrollado por Tony Hoare en 1960, es actualmente uno de los algoritmos de ordenación más eficientes y rápidos.

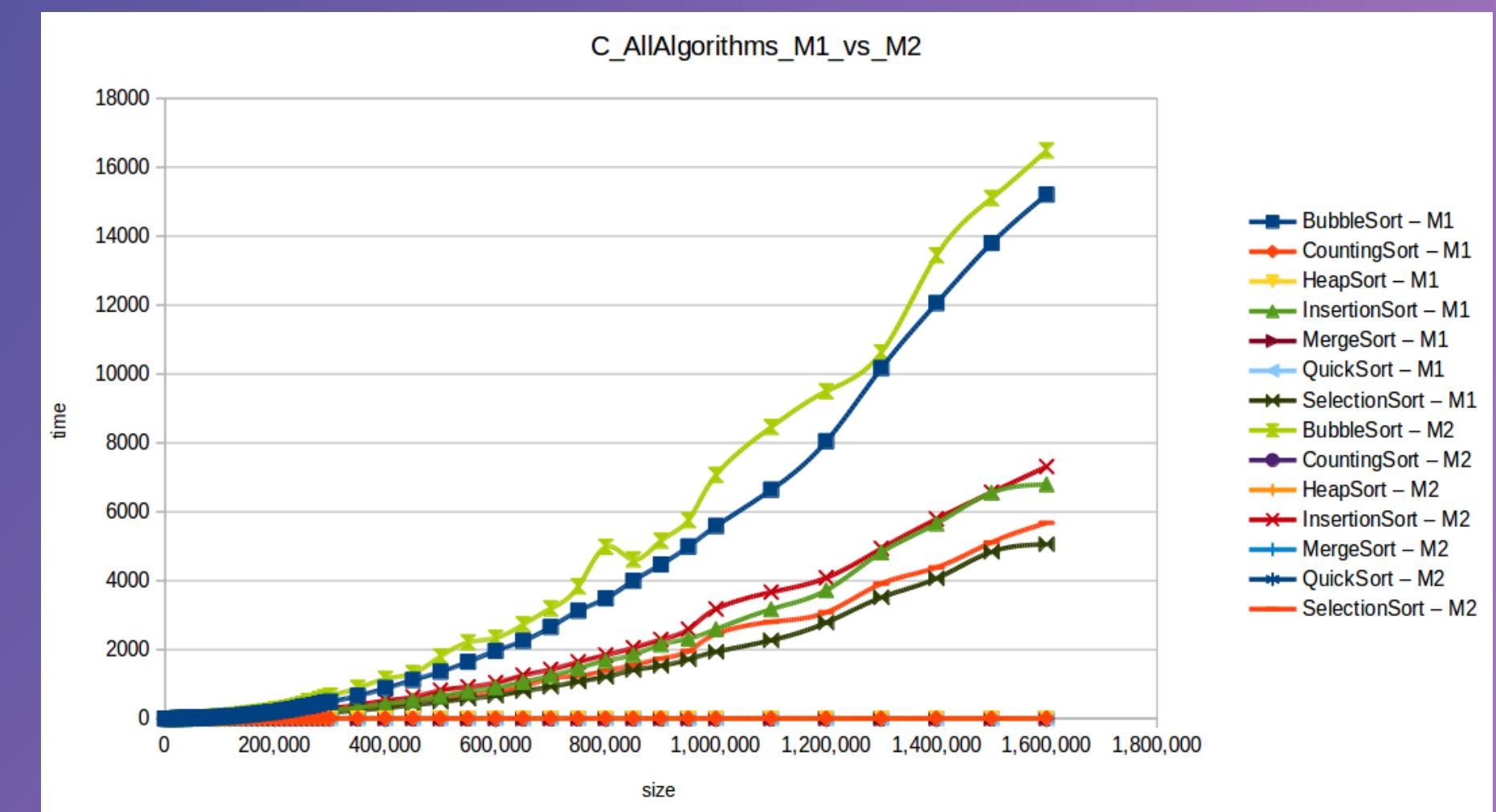
Este método también es conocido como método de ordenamiento por partición.



INTRODUCCIÓN

Basado en la técnica de divide y vencerás, permite ordenar (n) elementos en un tiempo promedio proporcional a: $O(n * \log(n))$

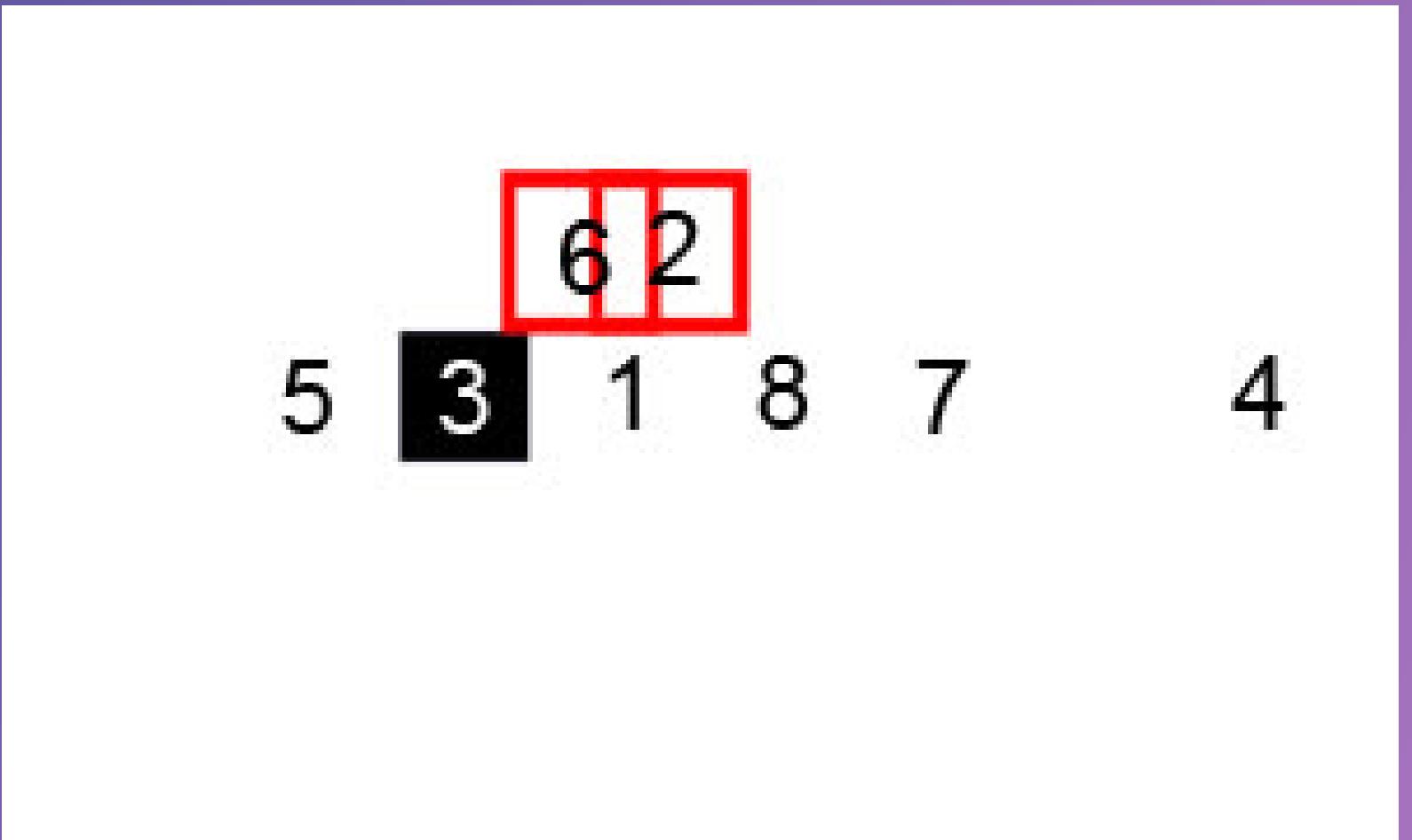
Lo que lo convierte en una mejora sustancial del método de intercambio directo y en el algoritmo de ordenamiento más utilizado a nivel mundial.



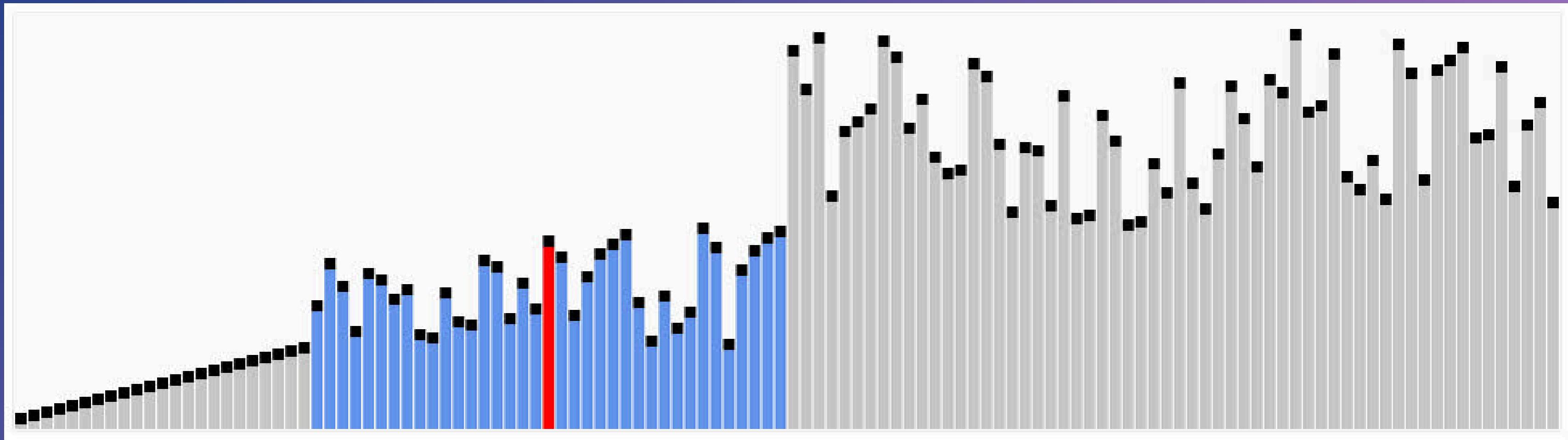
¿EN QUÉ CONSISTE?

Divide un conjunto de elementos en dos sub-listas mediante la elección de un pivote (punto de comparacion).

Los elementos menores o iguales al pivote se colocan en la primera sub-lista, mientras que los mayores se ubican en la segunda. Luego, el algoritmo se aplica de nuevo a las sub-listas, reorganizándolas hasta que toda la lista esté ordenada.



¿EN QUÉ CONSISTE?



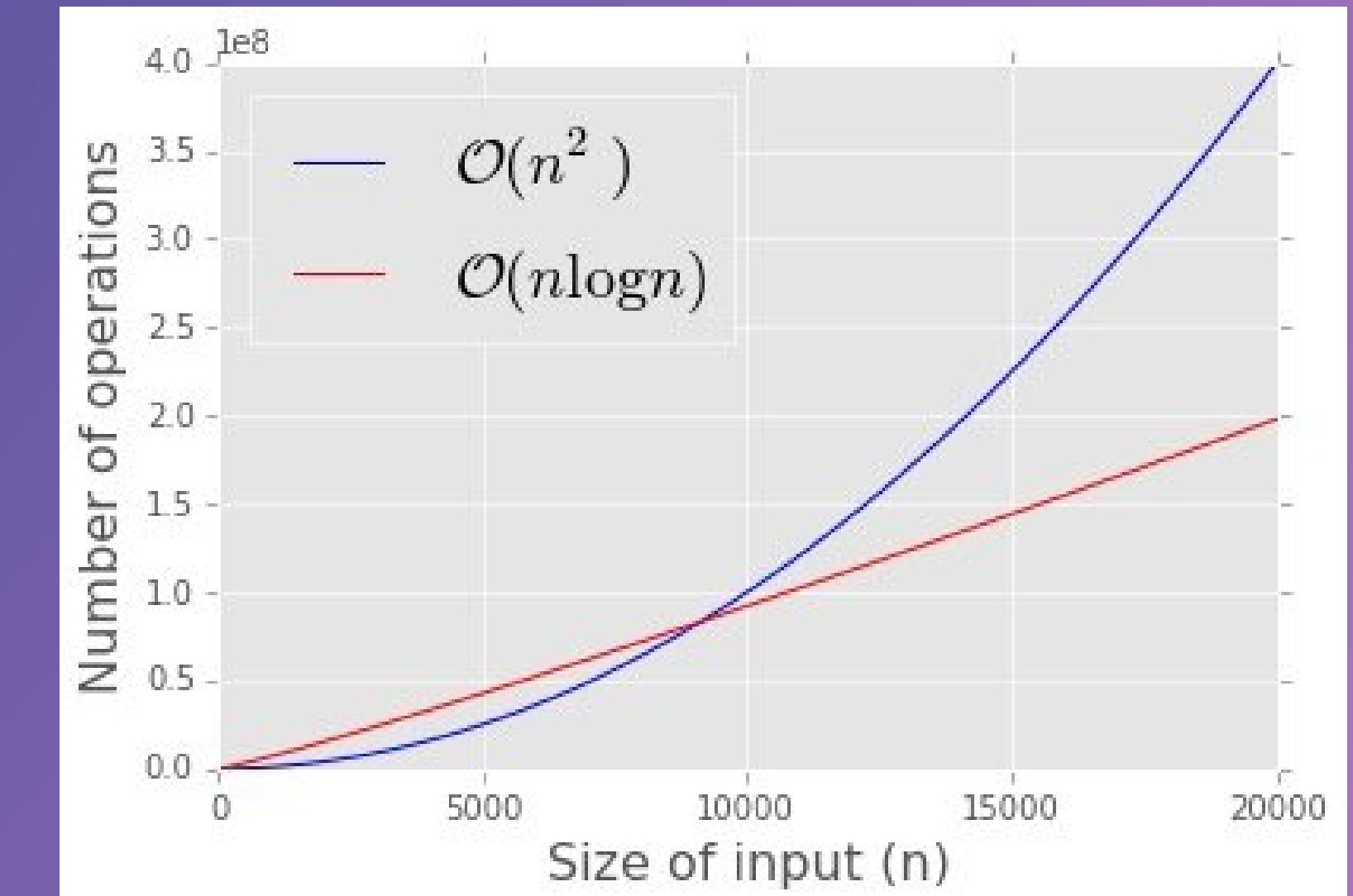
MEJOR / PEOR CASO

El mejor caso de Quicksort ocurre cuando el pivote divide la lista de manera equitativa, lo que lleva a una complejidad temporal de estable.

$O(n \log n)$

El peor caso es cuando el pivote es constantemente el valor extremo del arreglo, Como cuando la lista ya está ordenada, la complejidad aumenta.

$O(n^2)$



¿EN DONDE SE USA?

Quicksort se utiliza donde se tiene grandes volúmenes de datos. Es especialmente adecuado para aplicaciones donde la velocidad es crucial y no se necesita mantener el orden relativo de los elementos iguales.

Es ideal para arreglos que pueden dividirse fácilmente en listas más pequeñas, como en bases de datos, sistemas operativos y bibliotecas de software. También es útil en algoritmos que se benefician de una ordenación en el lugar, es decir, sin utilizar memoria adicional significativa.



VENTAJAS

QuickSort es un algoritmo de ordenamiento muy eficiente y rápido, especialmente en el caso promedio, donde su complejidad temporal es $O(n \log(n))$.

Esto se debe a que permite dividir el problema en sub-problemas más pequeños y manejables e ir juntando las soluciones hasta tenerla completa. Además, QuickSort es un algoritmo "in-place", lo que significa que no requiere espacio adicional significativo en memoria, haciendo un uso eficiente de los recursos disponibles.



DESVENTAJAS

En su peor caso, la complejidad temporal puede degradarse a $O(n^2)$. Esto sucede frecuentemente cuando los datos ya están ordenados o inversamente ordenados, lo que lo hace menos eficiente en estos escenarios.

QuickSort no es un algoritmo estable en su implementación básica, lo que significa que no necesariamente mantiene el orden relativo de los elementos iguales.

ITERACIONES

En total se hacen n iteraciones, debido a que todos los elementos son pivote alguna vez, y lo son máximo una vez.

Aunque muchas veces el elemento es pivote cuando es un solo valor (el mismo) por ordenar, entonces no se hace nada.

PREGUNTAS

1. ¿Cuál es la complejidad normalmente en un arreglo por el método QuickSort?



PREGUNTAS

1. ¿Cuál es la complejidad normalmente en un arreglo por el método QuickSort?

$O(n * \log(n))$



PREGUNTAS

2. ¿Qué es un pivote?



PREGUNTAS

2. ¿Qué es un pivote?

El pivote es un elemento seleccionado que se usa como punto de referencia.



PREGUNTAS

3. ¿Para qué tipo de arreglos funciona mejor cuando su complejidad es poca?



PREGUNTAS

3. ¿Para qué tipo de arreglos funciona mejor cuando su complejidad es poca?

Para arreglos con un gran volumen



PREGUNTAS

4. ¿Cuál es una de las ventajas del método QuickSort?



PREGUNTAS

4. ¿Cuál es una de las ventajas del método QuickSort?

No necesita de almacenamiento extra/Es muy rápido y eficiente



PREGUNTAS

5. ¿Con qué pivote se obtiene el caso más eficiente?



PREGUNTAS

5. ¿Con qué pivote se obtiene el caso más eficiente?

El valor del medio (tomando los valores, no la posición)



**GRACIAS POR SU
ATENCIÓN**