

Medición experimental de tiempos de ejecución en programas

Adrian Quiroga Linares

Abril de 2024

1 Introducción

El propósito de este informe es presentar una comparativa de los tiempos de ejecución de diversos algoritmos de ordenación, específicamente Quicksort, Bubblesort y Selección. Se analizará el rendimiento de estos algoritmos en relación con el tamaño de los vectores que serán ordenados y su correspondiente tiempo de ejecución.

2 Método Experimental

En el estudio, implementé el código de cada algoritmo en lenguaje C y utilicé la librería `vectordinámico.c`, adaptada para generar vectores grandes con valores aleatorios. También usé la librería `time.h` para medir los tiempos de ejecución de cada algoritmo, permitiendo establecer una relación entre el tamaño del vector y el tiempo de ejecución. Consideré un tiempo de ejecución razonable de 4 minutos, aunque es importante tener en cuenta que estos tiempos pueden variar según el hardware de cada ordenador.

3 Quicksort

Quicksort tiene una complejidad temporal de $O(n \log n)$. Utiliza la estrategia de 'divide y vencerás' y se basa en los siguientes pasos: selecciona un elemento como pivote de la lista, reordena la lista de modo que los elementos menores que el pivote queden antes, y los mayores después, y aplica recursivamente el mismo proceso a las sublistas generadas por la partición.



Figura 1: Esta imagen muestra el rendimiento del algoritmo Quicksort al ordenar vectores de diferentes tamaños. Podemos ver que incluso para tamaños de vector muy grandes, como un billón, el algoritmo es capaz de ordenar en aproximadamente cuatro minutos. Cabe destacar que esta gráfica está en escala logarítmica.

4 Bubblesort

El algoritmo Bubblesort es un método de ordenación que recorre repetidamente una lista de elementos, comparando cada par adyacente e intercambiándolos si están en el orden incorrecto. Esto se repite hasta que no realiza ningún intercambio. Tiene complejidad temporal de $O(n^2)$.

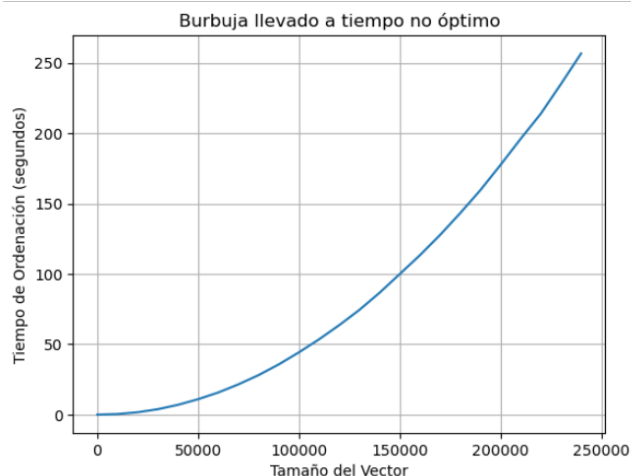


Figura 2: Esta imagen muestra el rendimiento del algoritmo Bubblesort al ordenar vectores de diferentes tamaños, hasta los 4 minutos.

5 Selección

El algoritmo de Selección es un algoritmo de ordenación que busca el mínimo elemento desde la posición actual y lo coloca en la posición correcta. Tiene complejidad temporal de $O(n^2)$.

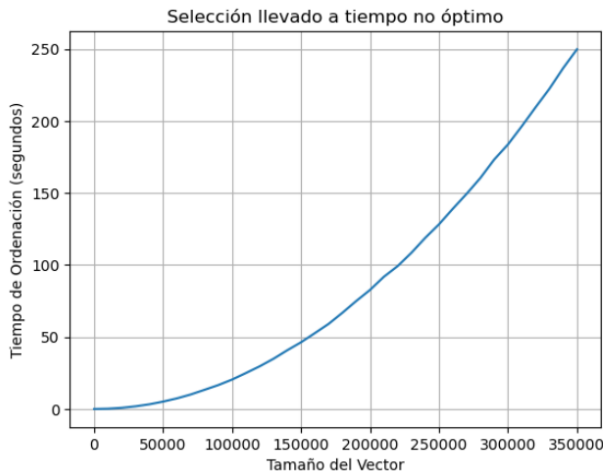


Figura 3: Esta imagen muestra el rendimiento del algoritmo de Selección al ordenar vectores de diferentes tamaños, hasta los 4 minutos.

6 Comparativa entre algoritmos

En este apartado realizo una comparativa de los 3 algoritmos para ver cuál es más eficiente y en qué casos:

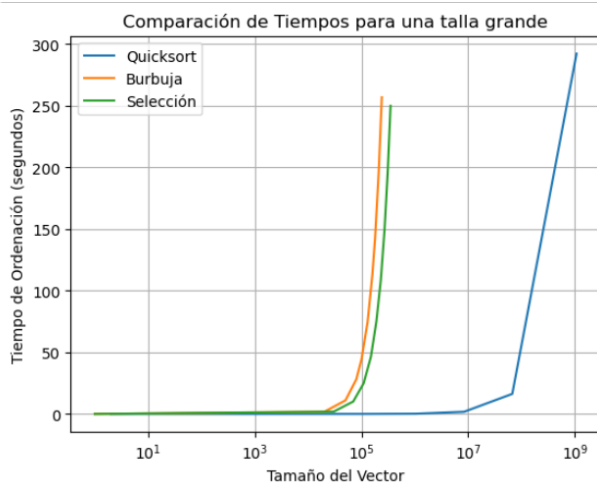


Figura 4: Esta imagen muestra la comparativa entre los tres algoritmos para tallas muy grandes, y podemos apreciar que tanto el método de selección como el de burbuja no pueden competir contra quicksort. Cabe destacar que está en escala logarítmica.

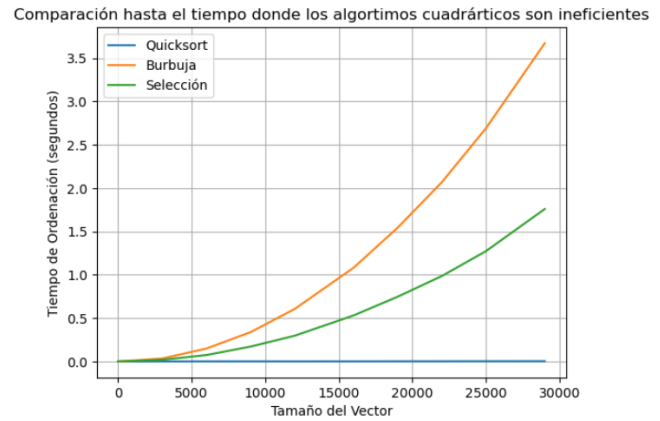


Figura 5: Esta imagen muestra con mayor detalle lo que sucedía en la anterior, podemos ver que tanto burbuja como selección llegan al tiempo no óptimo y quicksort lo hace mucho más rápido. Sin embargo ya podemos intuir que selección es mejor que burbuja.

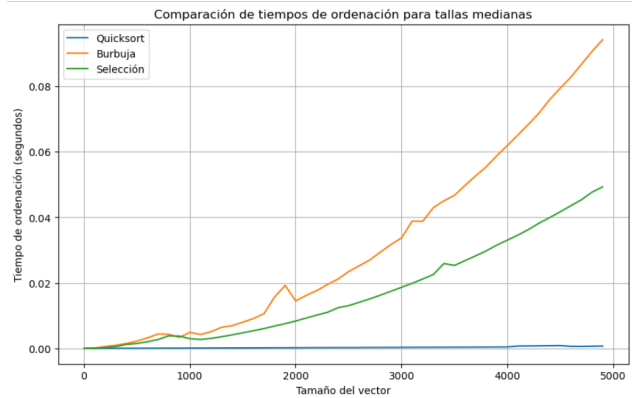


Figura 6: Reducimos el tamaño de la talla un poco más y se da la misma situación.

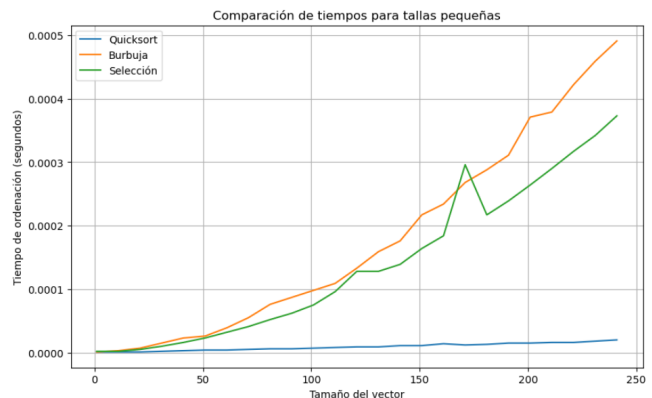


Figura 7: Al reducir la talla aún más se repite la misma situación, y tanto burbuja como selección se van acercando la una a la otra. Esos pequeños picos que aparecen en las gráficas son producidos por datos atípicos, puesto que al generar vectores aleatorios, algunos pueden estar más ordenados que otros y viceversa.

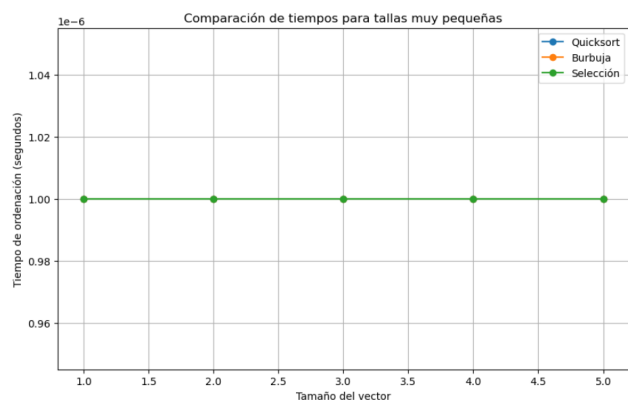


Figura 8: En esta imagen podemos apreciar que al usar una talla muy pequeña no hay diferencia. Sin embargo, no podemos afirmar que sean todos exactamente iguales, debido al método usado para la medición de tiempos.

pos. Si contásemos con una mayor precisión podríamos dar una respuesta sólida.

7 Conclusiones

Tras el estudio del funcionamiento de los 3 algoritmos podemos afirmar que Quicksort es más eficiente que Selección y Burbuja. Además Selección es más eficiente que Burbuja. Sin embargo, para tallas extremadamente pequeñas no podemos afirmar cuál es mejor debido a la precisión del método empleado para la medición de los tiempos.

8 Referencias

Aula virtual Programación II