



Instituto Politecnico Nacional

Unidad Profesional Interdisciplinaria de
Ingenieria Campus Zacatecas

19 de septiembre de 2023

Ingenieria en Sistemas Computacionales

Practica 1:Analisis de casos

Maestro:

M. en C. Erika Sanchez Femat

Materia:

Analisis y Diseño de Algoritmos

Alumno:

Alejandro Ulloa Reyes

Grupo:

3CM2

1. Introducción

Un algoritmo de ordenamiento burbuja consta de un arreglo que se recomienda no tenga un orden ascendente para que este pueda ordenarlo, es decir si se tiene un arreglo de la forma [14,5,27,2,9,10,3] el ordenamiento burbuja lo que hará es recorrer el arreglo y comparara si el primer elemento es menor que el segundo y de ser así lo cambiara de lugar, sino lo dejara como esta y así con cada elemento hasta que quede de manera ordenada [2,3,5,9,10,14,27].

2. Desarrollo

2.1. Análisis de casos

A continuación, se explicará cuáles son los casos mejores, peores y promedio de los algoritmos de ordenamiento burbuja y burbuja dinámica. Para el mejor caso del código de ordenamiento burbuja el mejor caso que podemos encontrar es cuando nuestro arreglo se encuentra acomodado de manera ordenada ya que el iterador solamente dará una vuelta para comprobar que cada elemento esta en su lugar correspondiente. Por lo tanto, el mejor caso tiene una complejidad de tiempo de $O(n)$ en la escala Big O, donde n es el número de elementos en el arreglo. En el peor caso todos los elementos del arreglo están acomodados de manera descendente, por lo que el iterador tendrá que pasar el último elemento hasta en frente y volver a comenzar a recorrer el arreglo hasta que quede completamente ordenado, esto resulta bastante ineficiente ya que en arreglos muy grandes tendrá que recorrer n cantidad de veces lo que lo convierte en $O(n^2)$ en la escala de dificultad de Big O. Para el caso promedio el arreglo estará desordenado de manera aleatoria con números que puedan ser sucesivos uno de otro en un par de ocasiones y después vayan variando según las órdenes del usuario lo que nos lleva a concluir que en la escala de complejidad Big O este caso se trata de $O(n^2)$ por lo mismo de que tendrá que recorrer el arreglo n cantidad de veces hasta que quede ordenado debidamente. En el mejor caso del ordenamiento burbuja mejorado es exactamente igual al ordenamiento por burbuja normal solo pasaría por cada elemento del arreglo para verificar que cada numero este ordenado. Por lo tanto, se trata de un $O(n)$. En cuanto al caso promedio y al peor su complejidad en Big O es la misma ($O(n^2)$) ya que los problemas son los mismos que en el algoritmo de burbuja simple ya que repasaran el arreglo n veces hasta que cumpla su propósito.

2.2. Comparación de resultados

Para el tiempo que tomo cada código en correr un arreglo de 5 elementos existe una leve diferencia entre uno y otro. Empezando con el código de ordenamiento burbuja simple, tome el cada caso y se corrió 5 veces. En el caso promedio se tomó un arreglo con los siguientes elementos [8,9,6,4,2], y se obtuvieron los siguientes tiempos.

Compilación No.	Tiempo
1	7.21s
2	10.67s
3	6.20s
4	7.18s
5	6.53s
Promedio.	7.558s

Cuadro 1: Mejores casos del ordenamiento Burbuja

Compilacion No.	Tiempo
1	6.90s
2	7.86s
3	5.61s
4	5.76s
5	6.94s
Promedio.	6.614s

Cuadro 2: Peores casos del ordenamiento Burbuja

Compilacion No.	Tiempo
1	14.55s
2	5.69s
3	9.90s
4	11.48s
5	7.43s
Promedio.	9.81s

Cuadro 3: Casos promedio del ordenamiento Burbuja

Compilacion No.	Tiempo
1	9.74s
2	5.84s
3	7.34s
4	5.87s
5	6.81s
Promedio.	7.12s

Cuadro 4: Mejores casos del ordenamiento de Burbuja dinamica

Compilacion No.	Tiempo
1	5.90s
2	6.12s
3	6.81s
4	6.60s
5	6.35s
Promedio.	6.35s

Cuadro 5: Peores casos del ordenamiento de Burbuja dinamica

Compilacion No.	Tiempo
1	7.04s
2	6.55s
3	7.38s
4	6.63s
5	7.51s
Promedio.	7.022s

Cuadro 6: Casos promedio del ordenamiento de Burbuja dinamica

3. Conclusiones

Finalmente, como conclusión en cuanto a su complejidad en Big O un algoritmo de otro no hay mucha o ninguna diferencia ya que cumplen con la misma función de maneras similares. Lo que me gustaría destacar es la diferencia de tiempos y sus variaciones ya que en algunas ocasiones el tiempo se reducía debido a que el usuario tecleaba de manera más rápida los elementos del arreglo, pero tratando de igualar el tecleo habitual los tiempos solamente tenían una varianza de milisegundos y en otras ocasiones el código corría mucho más lento tomando un par de segundos más de lo habitual para finalizar.

Referencias

<https://runestone.academy/ns/books/published/pythoned/SortSearch/ElOrdenamientoBurbuja.html>

<https://pablocianes.com/notacion-big-o/>

<https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx>