



**INSTITUTO POLITÉCNICO NACIONAL**  
Unidad Profesional Interdisciplinaria de Ingeniería  
Campus Zacatecas.

**MATERIA:**  
Análisis y Diseño de Algoritmos

**DOCENTE:**  
Erika Sanchez Femat

**PRACTICA:**  
○ Análisis de Casos

**ALUMNO:**  
○ Fernando Antonio García Ruiz

## **INTRODUCCION**

El método de ordenamiento burbuja, también conocido como "bubble sort" en inglés, es un algoritmo de ordenamiento simple pero ineficiente. Se utiliza para ordenar una lista de elementos, como números, en un orden específico, ya sea ascendente o descendente. Existen dos variantes principales del método burbuja: el burbuja normal y el burbuja optimizado.

## **MÉTODO DE ORDENAMIENTO DE BURBUJA**

El método de ordenamiento burbuja es un algoritmo de ordenamiento simple que funciona comparando repetidamente pares de elementos adyacentes en una lista y intercambiándolos si están en el orden incorrecto. El proceso se repite hasta que la lista esté completamente ordenada. Este algoritmo se llama "burbuja" porque los elementos más grandes "suben" gradualmente hacia la posición correcta, como burbujas en un vaso de refresco.

## **METODO DE ORDENAMIENTO DE BURBUJA OPTIMIZADA**

El método de ordenamiento burbuja puede ser optimizado para reducir la cantidad de comparaciones y, por lo tanto, mejorar su rendimiento. La optimización más común se conoce como "Optimización de Burbuja" o "Bubble Sort Optimizado". Esta optimización se basa en el hecho de que, después de una pasada completa a través de la lista, el elemento más grande ya está en su posición correcta en la última posición de la lista, por lo que no es necesario seguir comparándolo en las pasadas posteriores.

## ANALISIS DE CASOS

### Metodo de Ordenamiento de burbuja

Se comienza con una lista de elementos no ordenados

9 4 7 5 2

Toma los priemros dos números y si no estan ordenados se intercambiaran los lugares

4 9 7 5 2

Se repite el proceso hasta obtener el resultado final recorriendo el 9 en este ejemplo

4 7 9 5 2

4 7 5 9 2

4 7 5 2 9

Se vuelve a repetir el proceso desde el principio, pero debido a que el ultimo numero ya esta ordenado, ya no se evaluan el

ultimo valor, puesto que, el cuatro es menor que el siete, no hace el cambio y pasa al siguiente numero que es el siete y empieza a hacer el cambio

4 5 7 2 9

4 5 2 7 9

Se repite el proceso, ya que el cuatro es menor a cinco, se hace evalua el cinco con el dos, para despues, hacer el cambio de los valores cuatro y dos como se muestra a continuacion:

4 2 5 7 9

2 4 5 7 9

## **METODO DE ORDENAMIENTO DE BURBUJA OPTIMIZADA**

Contemplando lo del metodo anterior, se hace un proceso similar

## **PASO 1**

45 17 23 67 21 Se genera cambio en 45 y 17

17 45 23 67 21 Se genera cambio en 45 y 23

17 23 45 67 21 No hay cambio en 45 y 67

17 23 45 67 21 Se genera cambio en 67 y 21

17 23 45 21 67 Fin de la primera interacion

## **PASO 2**

17 23 45 21 67 No hay cambio

17 23 45 21 67 No hay cambio

17 23 45 21 67 Se genera cambio en 45 y 21

17 23 21 45 67 Fin de la segunda interaccion

## **PASO 3**

17 23 21 45 67 No hay cambio

17 23 21 45 67 Se genera cambio en 23 y 21

17 21 23 45 67 Fin de la tercera interacion

## PASO 4

17 21 23 45 67 No hay cambio

17 21 23 45 67 Fin de la cuarta iteraccion

## METODO DE ORDENAMINETO DE BURBUJA

**Mejor caso:** El mejor caso ocurre cuando: La lista ya está ordenada. En este escenario, el algoritmo de burbuja solo necesita realizar una pasada completa a través de la lista sin realizar ningún intercambio, ya que los elementos están en orden. Esto significa que el número de comparaciones es mínimo. Complejidad temporal:  $O(n)$ .

**Caso promedio:** El caso promedio es más difícil de determinar con precisión debido a las variaciones en los datos de entrada. Sin embargo, se espera que el algoritmo de burbuja realice un número cuadrático de comparaciones y swaps en promedio. Complejidad temporal:  $O(n^2)$ .

**Peor caso:** El peor caso ocurre cuando la lista está ordenada en orden inverso, de manera que cada elemento necesita ser movido hasta la posición correcta al final de la lista. Complejidad temporal:  $O(n^2)$ .

## METODO DE ORDENAMINETO DE BURBUJA OPTIMIZADA

**Mejor caso:** El mejor caso ocurre cuando la lista ya está ordenada. En este escenario, el algoritmo de burbuja optimizada solo necesita realizar una pasada completa a través de la lista sin realizar ningún intercambio, ya que los elementos están en orden. Esto significa que el número de comparaciones es mínimo. Complejidad temporal:  $O(n)$ .

**Caso promedio:** El caso promedio es similar al caso promedio del algoritmo de burbuja clásico. Aunque la versión optimizada reduce el número de comparaciones, todavía tiene un rendimiento cuadrático en promedio. Complejidad temporal:  $O(n^2)$ .

**Peor caso:** El peor caso ocurre cuando la lista está ordenada en orden inverso, de manera que cada elemento necesita ser movido hasta la posición correcta al final de la lista. Complejidad temporal:  $O(n^2)$ .

## REFERENCIAS

<https://runestone.academy/ns/books/published/pythoned/SortSearch/ElOrdenamientoBurbuja.html#:~:text=El%20ordenamiento%20burbuja%20hace%20m%C3%BAltiple,el%20lugar%20al%20que%20pertenece.>

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro9/metodo\\_de\\_burbuja.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro9/metodo_de_burbuja.html)

<https://juncotic.com/ordenamiento-de-burbuja-algoritmos-de-ordenamiento/>

<https://tutospoo.jimdofree.com/tutoriales-java/m%C3%A9todos-de-ordenaci%C3%B3n/burbuja-optimizado/>

[https://issuu.com/ernestoalonsopestanajimenez/docs/metodo\\_burbuja\\_optimizado.pptx](https://issuu.com/ernestoalonsopestanajimenez/docs/metodo_burbuja_optimizado.pptx)

<https://patsoptimizado.wordpress.com/>