

Instrucciones:

Sube aquí en formato pdf o video tu ejercicio sobre Bubble Sort + Binary Search (Búsqueda Binaria), en donde expliques EN TUS PALABRAS su complejidad temporal y si es más rápido o no por sobre la Búsqueda Secuencial.

Respuesta:**BB+BB vs. Búsqueda Secuencial**

Para comparar la complejidad temporal de dos métodos de búsqueda de un elemento en un array o lista, se deben primero entender ambos. El primer método en nuestra comparación era la búsqueda secuencial. Como lo vimos en clase, es un algoritmo de complejidad lineal. En el peor de los casos, debe recorrer toda la lista. Es por eso que su complejidad es $O(n)$. Al desconocer el orden de los elementos del array, el algoritmo debe buscar en cada uno de los espacios. Esto fue discutido en clase.

El segundo es la combinación de la búsqueda binaria con el bubble sort. El algoritmo de búsqueda binaria, en el mejor de los casos, encuentra el elemento que busca en una operación si se encuentra en el centro de la lista. En este algoritmo es indispensable que la lista esté ordenada. Sin embargo, en el peor de los casos tiene una complejidad de $\log_2(n)$, pues debe dividir la lista hasta que solo quede un elemento, y en ese caso encontrar el elemento o determinar que no está en la lista.

Para poder comparar, también hay que sumar la complejidad operacional de bubble sort. Este algoritmo consiste en checar cada par para ver si está en orden; si no lo está, hace el cambio, y pasa al siguiente par. De esta manera, si está completamente en orden, solo debe hacer el recorrido entero sin cambios. La cantidad de pares en una lista es $n - 1$. Sin embargo, en el peor de los casos, cuando la lista está completamente invertida, debe recorrer primero $n - 1$ pares, luego $n - 2$, hasta llegar a 0. De manera que la complejidad de operaciones del algoritmo es la suma de $n - 1$ hasta 0. Esta es la suma de $n - 1$ números consecutivos. Esta suma notable se conoce como la suma de Gauss. Esta se denota como $(n(n + 1)) / 2$. Al pensarla como hasta $n - 1$ se convierte en $(n - 1)(n) / 2$, que es $(n^2 + n) / 2$. En notación Big O se toma solo el término de mayor tasa de crecimiento; es decir, la complejidad de Bubble Sort es $O(n^2)$.

En cuanto a la complejidad conjunta, es $\log_2(n) + n^2$. En notación Big O se toma solo el término de mayor tasa de crecimiento; es decir, la complejidad de Bubble Sort + Búsqueda Binaria es $O(n^2)$.

De esta manera podemos concluir que la búsqueda secuencial es menos costosa en tiempo computacional, viendo el número de operaciones, con complejidad $O(n)$.

