

Clasificación

Sorting

Analogía general: Ordenando libros en una biblioteca mágica

Imaginá que tenés una gran **mesa desordenada llena de libros**, cada uno con un número en la tapa (que representa el valor que queremos ordenar). Tu objetivo es ordenarlos de menor a mayor y colocarlos en un estante.

La forma en que decidís hacerlo depende de tu estrategia o de cuán mágica sea tu biblioteca. Cada algoritmo representa una técnica distinta para lograrlo.

Insertion Sort:

Analogía: Bibliotecario paciente

Imaginá que vas sacando los libros uno por uno de la mesa y los vas colocando en el estante de forma ordenada. Cada vez que agarrás un libro, lo comparás con los que ya pusiste y lo insertás justo donde corresponde.

En el último recorrido (i -ésimo) se inserta el elemento en el lugar correcto entre los ($i-1$) elementos anteriores, los cuales ya fueron ordenados

Útil para:

- Arreglos pequeños o casi ordenado
- Funciona construyendo progresivamente una subsecuencia ordenada

Va de izquierda a derecha

Se examina el elemento y lo compara con los elementos de su izquierda

Se inserta el elemento en la posición correcta del array

Peor caso:

- En el peor caso (arreglo en orden inverso), su complejidad es $O(N^2)$.
 - En el mejor caso (arreglo ya ordenado), el tiempo es $O(N)$.
 - El caso promedio también es $O(N^2)$.

Posición de la matriz	0	1	2	3	4	5
Estado inicial	8	5	9	2	6	3
Después de ordenar a[0..1]	5	8	9	2	6	3
Después de ordenar a[0..2]	5	8	9	2	6	3
Después de ordenar a[0..3]	2	5	8	9	6	3
Después de ordenar a[0..4]	2	5	6	8	9	3
Después de ordenar a[0..5]	2	3	5	6	8	9

Figura 8.3 Acción básica del algoritmo de ordenación por inserción (la parte sombreada está ordenada).

El bucle interior ("mientras") tiene distinto comportamiento según el orden del conjunto:

- Mejor caso: (conjunto ya ordenado): Solo evalúa la condición sin entrar al bucle, por lo que es $O(1)$ por iteración, resultando en $O(N)$ en total.
- Peor caso (conjunto en orden inverso): Ejecuta el bucle $N - i$ veces por iteración, resultando en una complejidad $O(N^2)$.
- Caso promedio (claves al azar): También tiene una complejidad $O(N^2)$.

Peor caso y caso promedio: $O(n^2)$

Mejor caso: $O(n)$, cuando la lista estaba ordenada al revés

Selection Sort:

Analogía: El bibliotecario indeciso

Mirás todos los libros en la mesa, encontrás el más pequeño, y lo ponés primero en el estante. Luego repetís: buscás el siguiente más chico, y así.

$O(n^2)$ en todos los casos

Pasos:

1. Se busca la menor clave del conjunto, se transfiere al área de salida y se reemplaza por un valor infinito.
2. Se repite el proceso, buscando la segunda menor clave.
3. Se continúa hasta haber seleccionado todas las claves.

Shellsort:

Analogía: El bibliotecario con zancos

Primero comparás libros que están separados por una gran distancia (por ejemplo, cada 5 libros), los ordenás, y luego vas reduciendo esa distancia hasta comparar solo libros adyacentes. Así el desorden se reduce gradualmente. Ahí harías un insertion sort normal

Versión mejorada del Insertion Sort. Ordena por intervalos decrecientes.

Lo dicho en el ejemplo, hay un gap que va decreciendo y cuando llega a 1 se hace un insertion sort.

Tiempo: Varía según la secuencia de intervalos (gap), entre $O(n \log^2 n)$ y $O(n^{1.5})$ o $1.25)$

Peor caso $O(n^2)$

Quicksort:

El bibliotecario con varita mágica:

Elegís un libro llamado "pivote", y con un hechizo dividís los libros: los más pequeños a un lado, los más grandes al otro. Luego repetís este hechizo recursivamente con los dos grupos.

Divide y conquista. Elige un pivote, separa elementos menores y mayores, y aplica recursivamente.

- Mejor/promedio: $O(n \log n)$
- Peor: $O(n^2)$ si se elige mal el pivote (aunque se puede evitar con técnicas)

Merge Sort:

Analogía: El bibliotecario clonador

Dividís la pila de libros en mitades hasta que cada pila tiene un solo libro. Luego empezás a unir las pilas ordenadamente, como si fueras fusionando pilas de libros ordenadas. Recordar el reel que me envió Manu, en donde una lista se va dividiendo en dos hasta que haya 1 elemento en cada lado, luego se vuelve a unir en orden de menor a mayor.

Divide y conquista. Divide la lista hasta listas de 1 elemento, luego las fusiona ordenadamente.

Siempre $O(n \log n)$, muy estable

Usa espacio adicional (no in-place)

Heap Sort:

Analogía: El bibliotecario constructor

Construís una montaña mágica (heap) con los libros donde el más grande siempre está en la cima. Lo sacás (poniéndolo al final), reacomodás la montaña, y repetís.

Ordenación

Luego de construir el heap, se procede a extraer el mínimo (o máximo) repetidamente:

Tiempo: $O(n \log n)$

In-place pero no estable

Bin/Bucket Sort:

Analogía: El bibliotecario con cubos

Tenés varios baldes numerados. Vas colocando cada libro en su balde correspondiente según su número. Luego vaciás los baldes en orden.

- Muy rápido si los datos están distribuidos uniformemente
- Tiempo: $O(n + k)$ (k = número de buckets)
- Requiere conocer previamente el rango de los datos
- Distribuir los elementos en cubos: $O(n)$
 - Ordenar cada lista: $O(m)$ (si hay pocos elementos por cubo o se usa inserción)
 - Concatenación: $O(m)$

Esto implica que si el número de cubos m es proporcional a n (y los datos están distribuidos uniformemente), el rendimiento puede ser lineal.

Bubble Sort:

Analogía: El bibliotecario burbujeante

Comparas libros de a pares uno al lado del otro, y si están mal ordenados los intercambias. Hacés esto tantas veces como sea necesario, y los libros más grandes van "flotando" al final como burbujas.

Repite pasadas comparando elementos adyacentes e intercambiándolos si están en el orden incorrecto.

- Siempre $O(n^2)$, aunque en el mejor caso puede ser $O(n)$

Radix Sort

Analogía: El bibliotecario lector de dígitos

Ordenás los libros primero por el dígito menos significativo (por ejemplo, unidades), luego por decenas, centenas, etc. Usás baldes para cada dígito.

No compara elementos directamente, sino que agrupa por dígitos.

- Tiempo: $O(n * d)$, donde d = número de dígitos
- Requiere datos enteros y de longitud acotada

Counting Sort

Analogía: El bibliotecario contador

Imaginá que tenés una pila de libros con números del 0 al 9 en la tapa (ninguno más grande ni con decimales). En lugar de comparar libros entre sí, hacés una tabla y contás cuántos libros hay con cada número. Luego simplemente los acomodás uno tras otro según esa tabla.

- Tiempo: **$O(n + k)$** , donde:
 - n = cantidad de elementos
 - k = rango de valores (ej. si los números van de 0 a 100, $k=101$)
- Espacio adicional: **$O(k)$** (usa un arreglo extra para los conteos)