

Programando en Python

Ideas clave

Un algoritmo es una secuencia de instrucciones cuyo objetivo es la resolución de un problema.

Cuando hablamos de la complejidad de un algoritmo, nos referimos en realidad a una métrica teórica que mide la eficiencia computacional de un algoritmo.

Para conocer la complejidad de los algoritmos podemos clasificarlos en grupos, los más comunes son:

- **Constante:** $O(1)$. Es la más sencilla y siempre presenta un tiempo de ejecución constante.
- **Lineal:** $O(x)$. El tiempo crece linealmente mientras crece los datos
- **Polinómico:** $O(x^c)$, $c > 0$. Son los algoritmos más comunes. Si tienen 2 bucles anidados sería un cuadrático.
- **Logarítmico:** $O(\log x)$. Indican que el tiempo es menor que el tamaño de los datos de entrada.
- **Enelogarítmico:** $O(n \log x)$. Tan bueno como el anterior.
- **Exponencial:** $O(c^x)$. Es uno de las peores complejidades algorítmicas. Sube demasiado a medida que crece los datos de entrada.

Las notaciones asintóticas son herramientas matemáticas para representar la complejidad temporal de los algoritmos para el análisis asintótico. Hay principalmente tres notaciones asintóticas:

- **Notación theta (notación Θ):** encierra la función desde arriba y desde abajo.
- **Notación omega (notación Ω):** representa el límite inferior del tiempo de ejecución de un algoritmo. Por lo tanto, proporciona la mejor complejidad de caso de un algoritmo.
- **Notación Big-O (notación O):** representa el límite superior del tiempo de ejecución de un algoritmo. Por lo tanto, da la complejidad del peor de los casos de un algoritmo.

Hay cuatro tipos de estructuras de recopilación de datos en el lenguaje de programación Python:

- **La lista (list);** es una colección ordenada y modificable. Permite miembros duplicados.
- **Tupla (tuple);** es una colección ordenada e inmutable. Permite miembros duplicados.
- **Conjuntos (Set);** es una colección que no está ordenada ni indexada. No hay miembros duplicados.
- **Diccionario (Dictionary);** es una colección desordenada, modificable e indexada. No hay miembros duplicados.

Existen dos tipos de algoritmos de búsqueda:

- **Algoritmo de búsqueda lineal:** es el algoritmo de búsqueda más simple. También se llama búsqueda secuencial porque, en este algoritmo, buscamos un elemento atravesando todo el array y comparando cada elemento con el elemento deseado para encontrar una coincidencia. Si se encuentra el elemento deseado, se devuelve el índice o ese elemento; de lo contrario, continuamos buscando hasta agotar el array. También podemos buscar múltiples apariciones de un elemento dentro de un array. Se utiliza principalmente para buscar elementos dentro de un array sin clasificar. No se usa prácticamente porque es mucho más lento que la búsqueda binaria.
- **Algoritmo de búsqueda binaria:** es el algoritmo de búsqueda más rápido. Necesita que se ordene el array. Se basa en el enfoque de dividir y conquistar en el que dividimos el array en dos mitades y luego comparamos el elemento que estamos buscando con el elemento del medio. Si el elemento del medio coincide, devolvemos el índice del elemento del medio; de lo contrario, nos movemos a la mitad izquierda y derecha según el valor del artículo.

Algoritmos de ordenamiento:

- **Algoritmo de ordenamiento por inserción:** es un algoritmo de fácil aplicación que permite el ordenamiento de una lista. Su funcionamiento consiste en el recorrido por la lista seleccionando en cada iteración un valor como clave y compararlo con el resto insertándolo en el lugar correspondiente.
- **Orden de selección:** similar al orden de inserción con una ligera diferencia. Este algoritmo también divide la matriz en subpartes ordenadas y no ordenadas.
- **Ordenamiento de burbuja:** itera sobre la lista de datos, comparando elementos en pares hasta que los elementos más grandes “burbujean” hasta el final de la lista y los más pequeños permanecen al principio.
- **Merge sort:** El algoritmo de ordenamiento por mezcla es un algoritmo de ordenamiento externo estable basado en la técnica divide y vencerás. La idea de los algoritmos de ordenación por mezcla es dividir la matriz por la mitad una y otra vez hasta que cada pieza tenga solo un elemento de longitud. Luego esos elementos se vuelven a juntar (mezclados) en orden de clasificación.