

Algoritmo Insertion Sort

Consuelo Jorquera - Fecha de entrega: 12 de Mayo, 2025

¿Qué es?

El algoritmo de Insertion Sort (en español, de ordenamiento por inserción) es uno de los muchos algoritmos existentes para Python. Este algoritmo, tal como indica su nombre, funciona para ordenar listas desordenadas, ordenar usuarios, mantener un orden deseado mientras se ingresan datos, entre otras tareas. El texto se ajusta automáticamente alrededor de la imagen colocada a la derecha.

El algoritmo de Insertion Sort (en español, de ordenamiento e inserción) es uno de los muchos algoritmos existentes para Python. Este algoritmo, tal como indica su nombre, funciona para ordenar listas desordenadas, ordenar usuarios, mantener un orden deseado mientras se ingresan datos, entre otras.

Algorithm 1 Insertion Sort

Require: A = list of numbers

Ensure: A sorted in sequential order

```
1: function INSERTIONSORT( $A$ )
2:   for  $i \leftarrow 1$  to  $|A| - 1$  do
3:     open-index  $\leftarrow i$ 
4:     temp  $\leftarrow A_i$ 
5:
6:     for  $j \leftarrow (i-1)$  to  $0$  do
7:       if temp <  $A_j$  then
8:          $A_{\text{open-index}} \leftarrow A_j$ 
9:         open-index  $\leftarrow$  open-index - 1
10:      else
11:        break
12:      end if
13:    end for
14:
15:     $A_{\text{open-index}} \leftarrow$  temp
16:  end for
17: end function
```

Pseudocódigo

¿Quién lo creó?

Es difícil encontrar un creador fijo para este algoritmo, más hay reportes de que Kunrad Zuse fué el primer usuario, en 1945, en implementar el algoritmo de Insertion Sort en un programa como un recurso primitivo para el lenguaje de alto nivel: Plankalül

Usos comunes

Si bien hoy en día no es el método más eficiente para ordenar listas de gran tamaño, es bastante útil para ciertos casos específicos, tales como:

Listas pequeñas -Listas casi ordenadas -Online Sorting -Sistemas con recursos limitados -Ser parte de un algoritmo más complejo

¿Como lo usé?

Usé este algoritmo en mi código para tomar el segundo elemento de la lista y con ayuda de un bucle *while* intercambiamos el número actual con el número anterior mientras que el número anterior sea más grande que el número actual, este proceso se hace una y otra vez hasta que la lista esté completamente ordenada.

Bibliografía

Insertion Sort

Algoritmos de Ordenamiento y Búsqueda en Python: Optimizando la Gestión de Datos