

Nava Reyes Carlos – 17212163

Métodos de ordenamiento interno

Los métodos de ordenamiento interno son arreglos unidimensionales, pero su uso puede extender a otros tipos de arreglos y estructura de datos. Es importante señalar que no dependen de arreglos auxiliares para su ordenación.

Los métodos de ordenamiento interno a su vez se pueden clasificar en dos tipos:

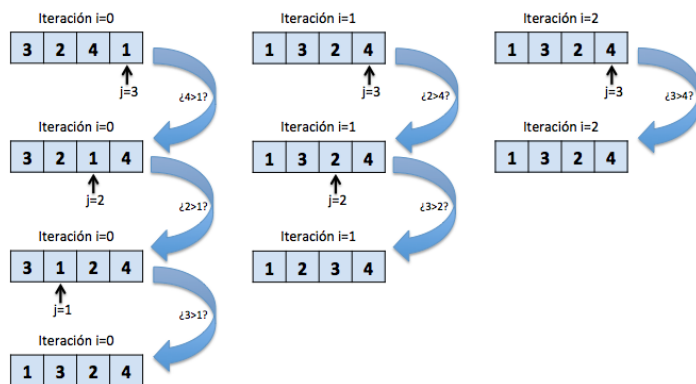
- Métodos directos (n^2)
- Métodos logarítmicos ($n \cdot \log n$)

Los métodos directos tienen la característica de que son fáciles de comprender, pero son ineficientes al trabajar con arreglos de tamaño mediano o superior.

Por otro lado, los métodos logarítmicos son más complejos, son más sofisticados de representar y son menos intuitivos. Sin embargo, son más eficientes ya que requieren de menos comparaciones y movimientos para ordenar los elementos.

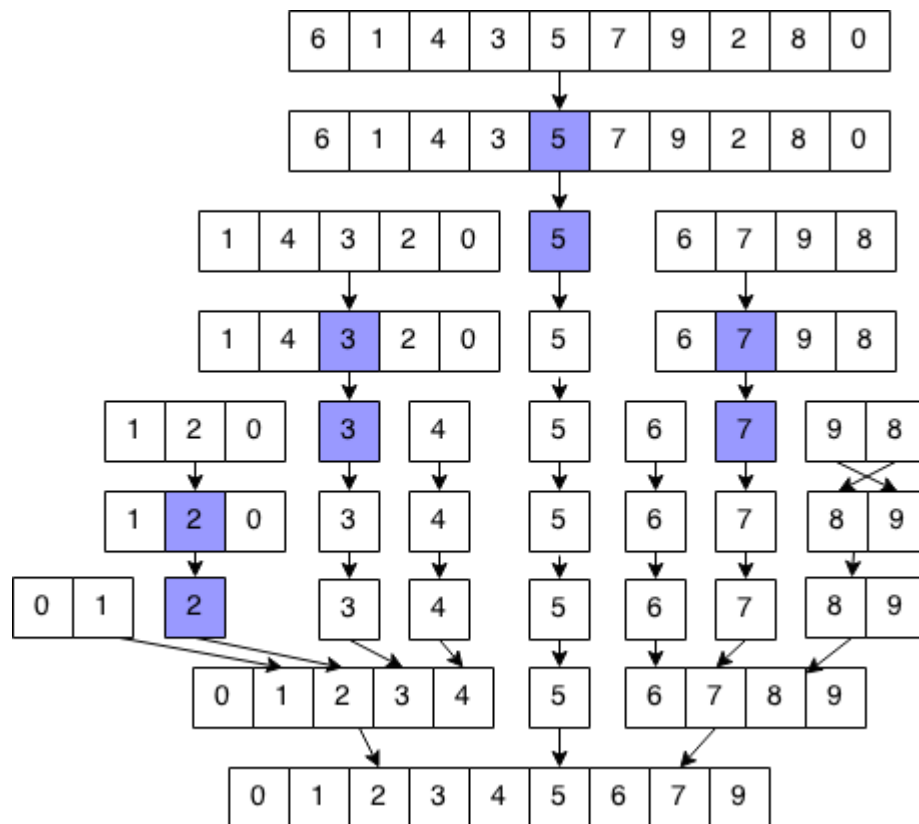
Método Burbuja:

El método de intercambio directo puede trabajar de dos maneras diferentes llevando los elementos más pequeños hacia la punta izquierda del arreglo o trasladando los elementos más grandes hacia su parte derecha. La idea básica de este algoritmo consiste en comparar pares de elementos adyacentes e intercambiarlos entre sí.



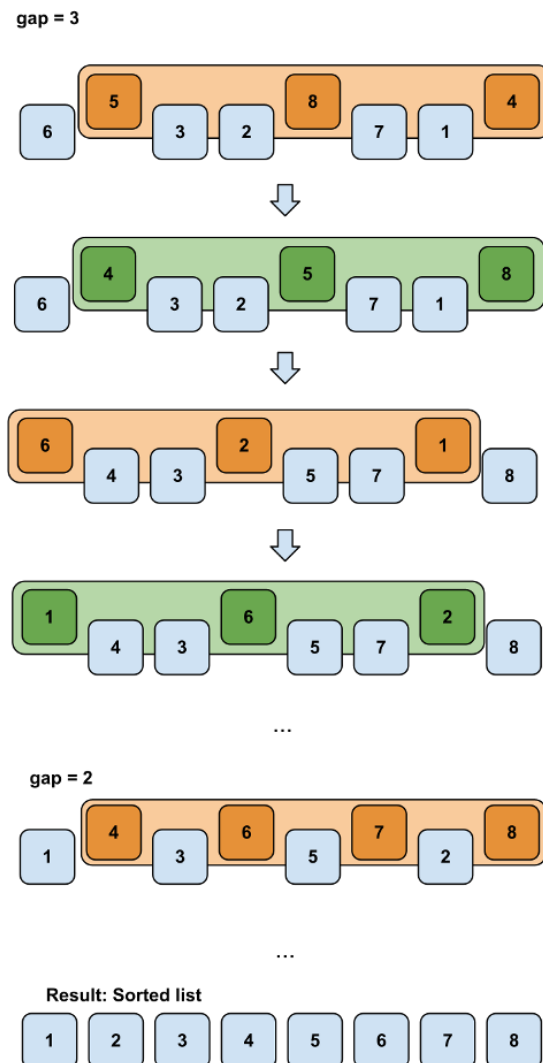
QuickSort:

Este método actualmente es el más eficiente y veloz de los métodos de ordenamiento interno. El ordenamiento rápido (quicksort en inglés) es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar n elementos en un tiempo proporcional a $n \log n$. Esta es la técnica de ordenamiento más rápida conocida. Fue desarrollada por C. Antony R. Hoare en 1960. El algoritmo original es recursivo, pero se utilizan versiones iterativas para mejorar su rendimiento (los algoritmos recursivos son en general más lentos que los iterativos, y consumen más recursos).



ShellShort:

El ordenamiento de Shell, a veces llamado “ordenamiento de incremento decreciente”, mejora el ordenamiento por inserción al romper la lista original en varias sublistas más pequeñas, cada una de las cuales se ordena mediante un ordenamiento por inserción. La manera única en que se eligen estas sublistas es la clave del ordenamiento de Shell. En lugar de dividir la lista en sublistas de ítems contiguos, el ordenamiento de Shell usa un incremento i , a veces denominado brecha, para crear una sublista eligiendo todos los ítems que están separados por i ítems.



Radix

Es un algoritmo de ordenamiento conocido en el mundo de la programación que ordena enteros a partir de sus dígitos de forma individual.

El siguiente ejemplo funciona con una lista desordenada de números.

1. En primer lugar, los va ordenando tomando en consideración el número menos significativo (la unidad) del más pequeño al más grande. (1ST PASS)
2. Luego, a partir de la lista que obtuvimos en el paso anterior, ordenamos los números de menor a mayor considerando esta vez la decena de cada uno de ellos. (2ND PASS)
3. Finalmente comprobamos que la lista fue ordenada satisfactoriamente mediante este procedimiento. (3RD PASS)

<u>INPUT</u>	<u>1st PASS</u>	<u>2nd PASS</u>	<u>3rd PASS</u>
10	10	5	5
52	52	209	10
5	44	10	19
209	5	19	44
19	209	44	52
44	19	52	209

Fuentes de información:

<http://interactivepython.org/runestone/static/pythoned/SortSearch/toctree.html>

Cairó, O., & Guarati, S. (2006). Estructuras de datos (3a. ed.). McGraw-Hill Interamericana.