**Método de ordenamiento por burbuja**

Es un sencillo algoritmo de ordenamiento que funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Para llevar a cabo el ordenamiento de burbuja, se lleva a cabo una operación básica, cambiando el elemento más grande, por el más pequeño que lo sigue.

**Desventaja:**

* Cuando la lista es muy grande, este método deja de ser eficiente.
* Consume bastante tiempo al momento de resolverlo una computadora.

**Ventaja:**

* Fácil de entender y muy práctico para ordenar listas hasta de 100

elementos.

* Es un código de ordenamiento reducido

**Método de ordenamiento por inserción**

La idea de este algoritmo de ordenación consiste en ir insertando un elemento de la lista ó un arreglo en la parte ordenada de la misma, asumiendo que el primer elemento es la parte ordenada, el algoritmo ira comparando un elemento de la parte desordenada de la lista con los elementos de la parte ordenada, insertando el elemento en la posición correcta dentro de la parte ordenada, y así sucesivamente hasta obtener la lista ordenada.

**Ventajas:**

* Es fácil de implementar
* Requiere poca memoria del sistema

**Desventajas:**

* Puede llegar a ser demorado en ordenar
* Realiza demasiadas comparaciones.

**Método de Ordenamiento Quickshort**

Su estrategia es la idea intuitiva de que es más fácil ordenar una gran estructura de datos subdividiéndolas en otras más pequeñas introduciendo un orden relativo entre ellas. En otras palabras, si dividimos el arreglo a ordenar en dos subarreglos de forma que los elementos del subarreglo inferior sean más pequeños que los del subarreglo superior, y aplicamos el método reiteradamente, al final tendremos el arreglo inicial totalmente ordenado.

La idea central de este algoritmo consiste en los siguiente: Se toma un elemento x de una posición cualquiera del arreglo. Se trata de ubicar a x en la posición correcta del arreglo, de tal forma que todos los elementos que se encuentran a su izquierda sean menores o iguales a x y todos los elementos que se encuentren a su derecha sean mayores o iguales a x. Se repiten los pasos anteriores pero ahora para los conjuntos de datos que se encuentran a la izquierda y a la derecha de la posición correcta de x en el arreglo

**Ventajas:**

* Muy rápido.
* No requiere memoria adicional.

**Desventajas:**

* Implementación un poco más complicada.
* Recursividad (utiliza muchos recursos).
* Mucha diferencia entre el peor y el mejor caso.