MÉTODOS DE ORDENACIÓN

ESTRUCTURAS DE DATOS

Profesor: Oscar Flores
Nombre: Daniela Beatriz Martínez Montes
Grupo 2.-B

08/04/2017

Métodos de ordenamiento

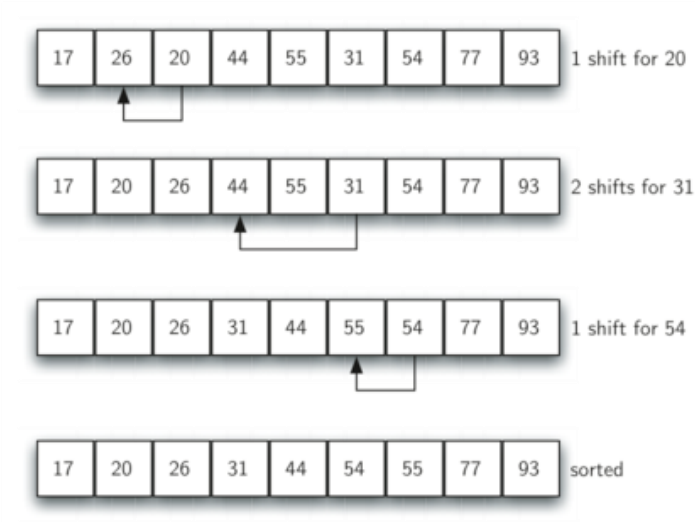
Método de ordenamiento Shell

Características

• Se trata de un algoritmo de ordenación interna.
• Se basa en comparaciones e intercambios.
• En cierto modo, puede considerarse una ampliación del algoritmo de inserción directa.
• No es estable: dados dos elementos que al compararlos sean "iguales".

Análisis y nivel de complejidad:

El algoritmo Shell mejora el ordenamiento por inserción comparando elementos separados por un espacio de varias posiciones. Esto permite que un elemento haga "pasos más grandes" hacia su posición esperada. El último paso del Shell es un simple ordenamiento por inserción, pero para entonces, ya está garantizado que los datos del vector están casi ordenados. Por todo esto, Shell es un algoritmo de ordenación interna muy sencillo pero muy ingenioso, basado en comparaciones e intercambios, y con unos resultados radicalmente mejores que los que se pueden obtener con el método de la burbuja, el de selección directa o el de inserción directa, ya que es mas preciso, por lo tanto no tiene demasiada complejidad



Conclusión

Es evidente que el ShellSort es un buen método de ordenamiento, además que es sencillo de comprender, solo tienes que comprender el ordenamiento por inserción y este método se te hará muy sencillo y eficaz

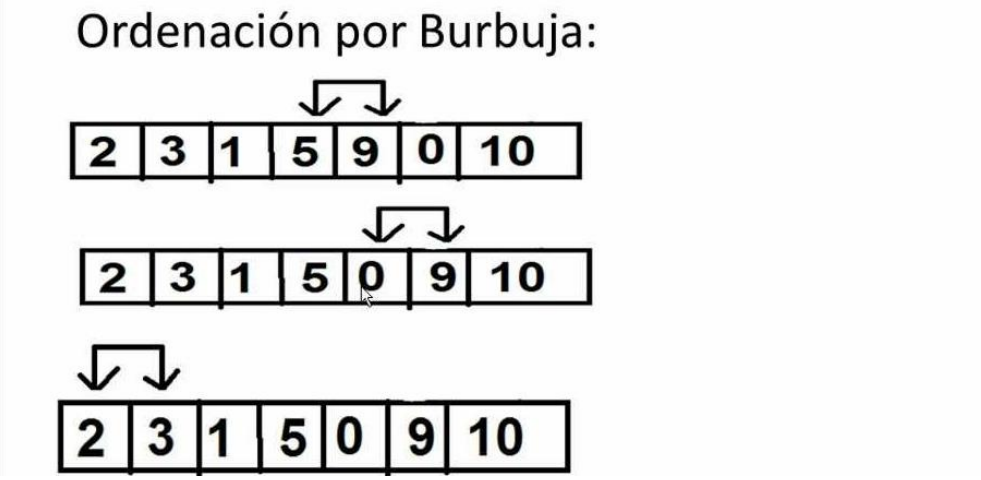
Método burbuja

Características

- Funciona comparando elementos de dos en dos en un ciclo, es decir compara los números adyacentes, intercambiándolos según sea el caso ascendente o descendente
- Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios

Análisis y nivel de complejidad

El método de la burbuja es uno de los más simples para aprender, es tan fácil como comparar todos los elementos de una lista contra todos, si se cumple que uno es mayor o menor a otro, entonces los intercambia de posición, el único problema es que tardas demasiado tiempo, ya que básicamente vas comparando numero por número y esto requiere muchas escrituras de memoria



Conclusión:

El método Burbuja podría ser excelente si se tiene un arreglo de pocos datos para el ordenamiento, pero si el arreglo tiene una gran infinidad de número, este método sería muy ineficiente, si bien es un método de los más fáciles de aprender, actualmente no es muy utilizado debido a su ineficacia para grandes cantidades de números

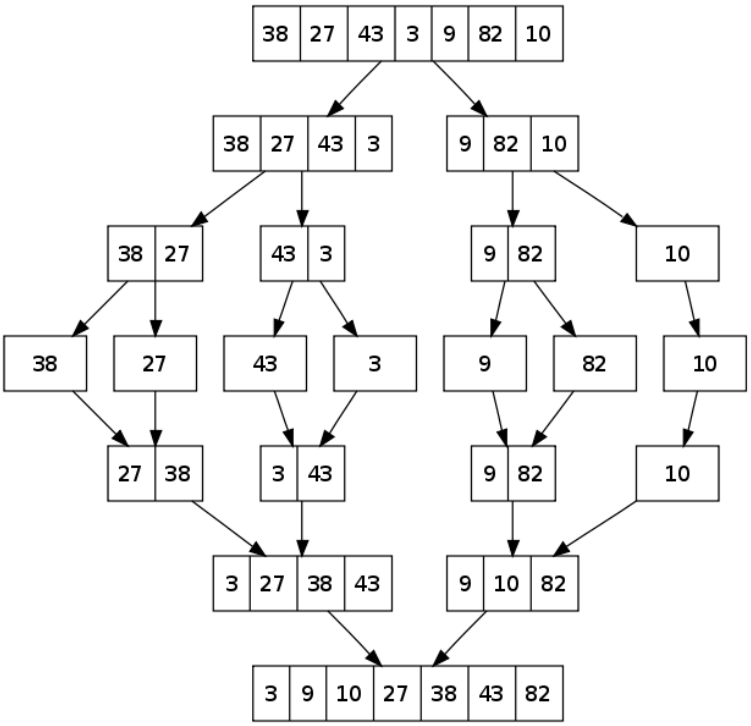
Metodo Merge Sort

Características

- Combina (intercala) dos estructuras previamente ordenadas.
- Dividir el grupo de datos en dos y ordena por separado cada mitad.
- Cuando se tengan las mitades ordenadas, pueden irse mezclando para obtener fácilmente una secuencia ordenada.

Análisis y nivel de complejidad:

La eficiencia de este algoritmo es bastante notable en tiempo de ejecución en comparación con otros, ya que su manera de trabajo por grupos pequeños agiliza la organización de los datos, pero a pesar de esto, el merge sort es bastante difícil de implementar y aprender para los principiantes, especialmente la parte de la fusión del algoritmo., ya que este es muy complejo



Conclusiones

Su utilización se da con mucha frecuencia cuando la cantidad de registros no es muy grande ya que para hacer las mezclas éste método utiliza el doble del espacio que gasta el arreglo original de valores, por lo tanto es otro método ineficiente, ya que tarda demasiado y además ocupa más memoria si está trabajando con grandes cantidades de datos, se considera que es aún más ineficiente que el de burbuja

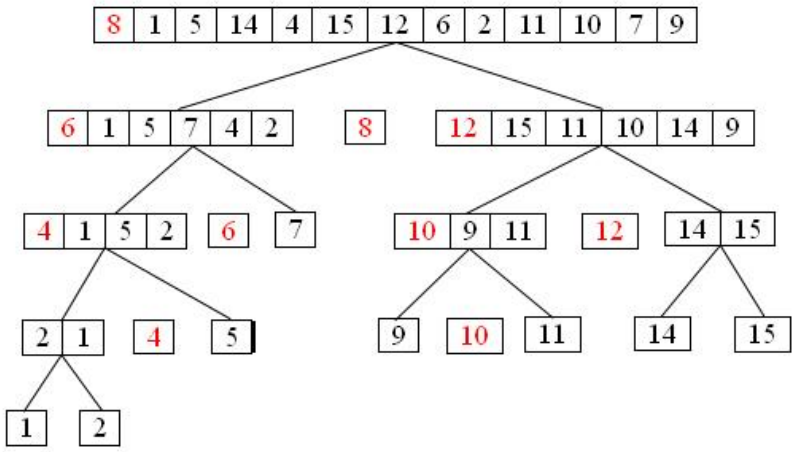
Método QuickSort

Características:

- Trabaja mejor para elementos de entrada desordenados completamente, que para elementos semiordenados.
- Este tipo de algoritmos se basa en la técnica "divide y vencerás", o sea es más rápido y fácil ordenar dos arreglos o listas de datos pequeños, que un arreglo o lista grande.
- Normalmente al inicio de la ordenación se escoge un elemento aproximadamente en la mitad del arreglo, así al empezar a ordenar, se debe llegar a que el arreglo este ordenado respecto al punto de división o la mitad del arreglo.
- Se podrá garantizar que los elementos a la izquierda de la mitad son los menores y los elementos a la derecha son los mayores.

Análisis y nivel de complejidad:

Este método no es para nada estable, su Implementación resulta un poco más complicada para principiantes además de que este método utiliza demasiados recursos



Juntando los elementos, el arreglo quedaría ordenado

1	2	4	5	6	7	8	9	10	11	12	14	15

Conclusiones

Se podría concluir que puede llegar a ser uno de los mejores métodos, ya que es uno de los más veloces que existen, y que puede llegar a ordenar gran cantidad de número en muy poco tiempo, pero la principal desventaja es que es un método que sigue utilizando muchos recursos y memoria

MÉTODO DE INSERCIÓN

El ordenamiento por inserción es una manera muy natural de ordenar para un ser humano, y puede usarse fácilmente para ordenar un mazo de cartas numeradas en forma arbitraria.

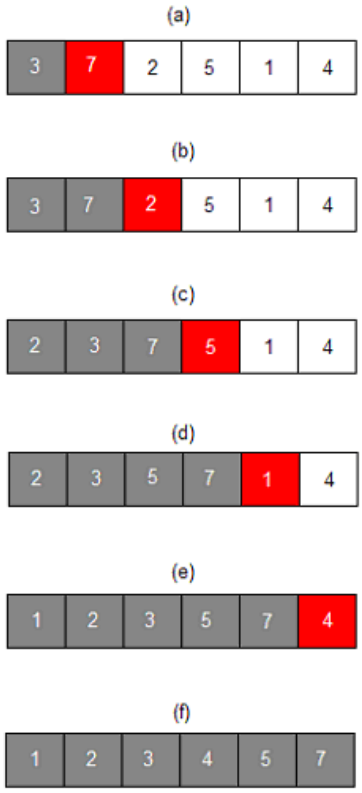
Inicialmente se tiene un solo elemento, que obviamente es un conjunto ordenado. Después, cuando hay *k* elementos ordenados de menor a mayor, se toma el elemento *k+1* y se compara con todos los elementos ya ordenados, deteniéndose cuando se encuentra un elemento menor (todos los elementos mayores han sido desplazados una posición a la derecha) o cuando ya no se encuentran elementos (todos los elementos fueron desplazados y este es el más pequeño). En este punto se *inserta* el elemento *k+1* debiendo desplazarse los demás elementos.

Análisis y nivel de complejidad

Para estimar la complejidad de este algoritmo, tomaremos como operaciones elementales para calcular f(n), las comparaciones y asignaciones que se realizan involucrando números a ordenar. El peor caso para obtener una ordenación ascendente del vector, se produce cuando los números están inicialmente ordenados de mayor a menor, es decir totalmente al revés. En este caso, el número de comparaciones necesarias es, en función del contador del algoritmo.

Conclusión

Este método de ordenamiento a pesar de que es muy sencillo de comprender y además de ser muy eficiente si la entrada de datos está “casi ordenada” de menor a mayor, tiene sus contras por el hecho de mover los valores sólo una posición cada vez.



MÉTODO SELECCIÓN

El ordenamiento por selección es un algoritmo de ordenamiento que requiere O (n^2) operaciones para ordenar una lista de n elementos.
Su funcionamiento es el siguiente:

- Buscar el mínimo elemento de la lista
- Intercambiarlo con el primero
- Buscar el mínimo en el resto de la lista
- Intercambiarlo con el segundo

Y en general:

- Buscar el mínimo elemento entre una posición *i* y el final de la lista
- Intercambiar el mínimo con el elemento de la posición *i*

Análisis y Nivel de complejidad

El ciclo externo se ejecuta n veces para una lista de n elementos, o sea que para ordenar un vector de n términos, tiene que realizar siempre el mismo número de comparaciones.
Cada búsqueda requiere comparar todos los elementos no clasificados, de manera que el número de comparaciones c(n) no depende del orden de los términos, si no del número de términos; por lo que este algoritmo presenta un comportamiento constante independiente del orden de los datos. Luego la complejidad es del orden n2.

Conclusión

Existe cierto tipo de desacuerdo en cuanto a si este algoritmo es estable o no, pero en realidad este método es bastante estable. Ya que es de fácil implementación, no requiere de una memoria adicional, no realiza gran cantidad de cambios, solo son la cantidad de datos que contenga el vector.Por otro lado en gran cantidad de datos es lento y poco eficiente ya que realiza grandes comparaciones.