

INTRODUCCIÓN

En este proyecto podrás ver los diferentes tipos de métodos de ordenamiento, abordaremos tanto la lógica de su estructura como en su eficacia, comparándolos unos con otros utilizando un mismo arreglo de 50 números y temporalizando lo para la comprobación la misma.

DESARROLLO:

A continuación se explicara cada método utilizado:

Método Bubble:

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. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada.

El procedimiento consiste en ir comparando desde la casilla 0 número tras número hasta encontrar uno mayor, si este es realmente el mayor de todo el vector se llevará hasta la última casilla, si no es así, será reemplazado por uno mayor que él. Este procedimiento seguirá así hasta que haya ordenado todas las casillas del vector. Una de las deficiencias del algoritmo es que ya cuando ha ordenado parte del vector vuelve a compararlo cuando esto ya no es necesario.

Método Quick-sort:

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.

El procedimiento consiste en elegir un elemento de la lista de elementos a ordenar, al que llamaremos pivote con el cual vamos a resituar los demás elementos de la lista a cada lado de él, de manera que a un lado queden todos los menores que él, y al otro los mayores. En este momento, el pivote ocupa exactamente el lugar que le corresponderá en la lista ordenada. La lista queda separada en dos sublistas, una formada por los elementos a la izquierda del pivote, y otra por los elementos a su derecha, tendremos que repetir este proceso de forma recursiva para cada sublista mientras éstas contengan más de un elemento.

Una vez terminado este proceso todos los elementos estarán ordenados. Como se puede suponer, la eficiencia del algoritmo depende de la posición en la que termine el pivote elegido.

Método Merge sort:

Este algoritmo consiste en dividir la lista a resolver en sublistas del mismo tipo que a su vez se dividirán, mientras no sean suficientemente pequeños.

El procedimiento consiste en ordenar una secuencia x de elementos, si F tiene uno o ningún elemento, está ordenada, si x tiene al menos dos elementos se divide en dos secuencias F_1 y F_2 , F_1 conteniendo los primeros $n/2$, y F_2 los restantes, seguimos a ordenar F_1 y F_2 , aplicando recursivamente este procedimiento. Mezclar S_1 y S_2 ordenadamente en F . Mezclar dos secuencias ordenadas F_1 y F_2 en F : Se tienen referencias al principio de cada una de las secuencias a mezclar (F_1 y F_2). Mientras en alguna secuencia queden elementos, se inserta en la secuencia resultante (F) el menor de los elementos referenciados y se avanza esa referencia una posición.

Método Shell sort:

Es un algoritmo similar a la del Bubble Sort, sin embargo no ordena elementos adyacentes sino que utiliza una segmentación entre los datos. Esta segmentación puede ser en cualquier tamaño de acuerdo a una secuencia de valores que empiezan con un valor grande y van disminuyendo hasta llegar al '1'.

Ordena subgrupos de elementos separados F unidades (respecto de su posición en el arreglo) del arreglo o lista original. El valor F es llamado incremento.

Después de que los primeros K subgrupos han sido ordenados (generalmente se utiliza inserción directa), se escoge un nuevo valor de F más pequeño, y el arreglo es de nuevo partido entre el nuevo conjunto de subgrupos. Cada uno de los subgrupos mayores es ordenado y el proceso se repite de nuevo con un valor más pequeño de F .

TABLA DE RESULTADOS

I	Bubblesort	Quicksort	Mergesort	Shellsort
1	0.00939298	0.00503826	0.00298715	0.00400519
2	0.0090127	0.00458121	0.00495696	0.00498343
3	0.00826335	0.00498772	0.00299072	0.00399184
4	0.0099473	0.00500655	0.00295162	0.00498867
5	0.00897646	0.0043447	0.00600171	0.00398707
6	0.0098505	0.00398731	0.00399709	0.0040307
7	0.00927043	0.00402188	0.00399041	0.00499153

8	0.01202631	0.00694442	0.00799513	0.00573683
9	0.00974607	0.00697207	0.00794435	0.00597191
10	0.00800991	0.00533342	0.00717664	0.00798297
11	0.00897598	0.00401115	0.00697899	0.00399375
12	0.00797772	0.00398922	0.00402498	0.00498867
13	0.01097512	0.00399351	0.00398922	0.0039897
14	0.01097417	0.00399208	0.00398803	0.00399232
15	0.00997376	0.00595665	0.00401926	0.00498271
16	0.00697398	0.00401545	0.00527692	0.00379443
17	0.00897789	0.00398874	0.00363922	0.00495648
18	0.01396537	0.00498629	0.00498581	0.00400114
19	0.00900221	0.00598669	0.00399923	0.00401783
20	0.00900674	0.0040226	0.00402546	0.00395083
21	0.00997543	0.00398684	0.00776839	0.00599217
22	0.00897312	0.00398946	0.00395989	0.00498366
23	0.01097012	0.00498438	0.00398302	0.00499487
24	0.00997448	0.00400043	0.00510383	0.00499988
25	0.00940609	0.00826812	0.00780368	0.00730968
26	0.00901055	0.00697994	0.00598431	0.00398707
27	0.01233435	0.0081532	0.00698113	0.00804543
28	0.00897598	0.00501204	0.00498509	0.00395226
29	0.00901079	0.00395656	0.00488567	0.00398898
30	0.00897455	0.00706339	0.00735164	0.00725818
promedio:	0.00963015	0.00508514	0.00515752	0.00496167

CONCLUSIÓN

Basándome en la tabla de resultados llegue a la conclusión que el método Shell sort fue el que mostró una mayor eficacia dando un resultado promedio de 0.00496167 segundos, por el lado contrario el método Bubble es clasificado como el menos eficaz debido a su alto promedio de 0.00963015 segundos.

Posicionamiento de eficacia:

- 1- Shellsort - 0.00496167 segundos
- 2- Quicksort - 0.00508514 segundos
- 3- Mergesort - 0.00515752 segundos
- 4- Bubble - 0.00963015 segundos