

## Ejercicio de clase 1 Díaz Hernández Marcos Bryan

1) ¿Cómo se define la operación de ordenamiento?

El ordenamiento para mí es dar un sentido a las cosas en base a un aspecto a destacar para que se de un orden para que los elementos destaquen dependiendo si es orden ascendente o descendente.

2) ¿Qué significa la "verificación" de un algoritmo de ordenamiento?

Sirve para verificar / bueno para comprobar si se han cumplido las funciones que tendría que hacer, es como darle el sello de calidad y que cumple con lo que dice hacer.

3) Además de facilitar la operación, ¿cuáles serían otros objetivos de la operación de ordenamiento?

Así como funciona para ordenar puede ayudar a colocar elementos nuevos y no crear desorden, además reducir la complejidad de los algoritmos y por consecuencia aprovechar los recursos de la computadora para procesos secundarios o el optimizar el uso de la memoria y de la velocidad del procesamiento.

4) En clase se mencionó la clasificación de algoritmos de ordenamiento internos y externos, investiga a qué se refiere la clasificación de "directos" e "indirectos".

Los algoritmos indirectos / directos tienen la característica de ser sencillos en la utilización pero fallan cuando la cantidad o el valor de los datos tiende a ser grande.

Por el contrario los indirectos toman el camino contrario y también se conocen como logarítmicos, por ser más complejos pero eficiente en la cantidad de datos que pueden procesar de forma continua.

5) Ordena los siguientes valores utilizando el algoritmo de insertion sort.

{Claves 9, 17, 46, 8, 23, 33, 44, 12}

1 2 3 4 5 6 7  
17 46 8 23 33 44 12  
1) 17 46 8 23 33 44 12  
2) 17 46 8 23 33 44 12  
3) { 17 46 46 23 " " "  
17 17 46 23 " " "  
8 17 46 23 " " "  
4) { 8 17 46 46 " " "  
8 17 23 46 " " "  
5) { 8 17 23 46 46 " "  
8 17 23 33 46 " "  
6) { 8 17 23 33 44 46 "  
8 17 23 33 44 46 46  
8 17 23 33 44 44 46  
7) { 8 17 23 33 33 44 46  
8 17 23 23 33 44 46  
8 17 17 23 33 44 46  
8 12 17 23 33 44 46

1)  $n=7$   
for  $i=1$  to 7  
index = 17  
 $j=0$   
while ( $0 > 0$  algo  $> 17$ )  
no ample  
• list[1] = 17

3)  $n=7$   $i=3$   
index = 8  
 $j=2$   
while ( $2 > 0$  y  $46 > 8$ )  
• list[5] = 46  
 $j=1$   
while ( $1 > 0$  y  $17 > 8$ )  
• list[2] = 17  
 $j=0$   
while ( $0 > 0$  y algo  $> 8$ )  
• list[1] = 8

5)  $n=7$   $i=5$   
index = 33  
 $j=4$   
while ( $4 > 0$  y  $46 > 33$ )  
• list[5] = 46  
 $j=3$   
while ( $3 > 0$  y  $23 > 33$ )  
• list[4] = 33

2)  $n=7$   $i=2$   
index = 46  
 $j=1$   
while ( $1 > 0$  y  $17 > 46$ )  
• list[2] = 46

4)  $n=7$   $i=4$   
index = 23  
 $j=3$   
while ( $3 > 0$  y  $46 > 23$ )  
• list[4] = 46  
 $j=2$   
while ( $2 > 0$  y  $17 > 23$ )  
• list[3] = 23

6)  $n=7$   $i=6$   
index = 44  
 $j=5$   
while ( $5 > 0$  y  $46 > 44$ )  
• list[6] = 46  
 $j=4$   
while ( $4 > 0$  y  $33 > 44$ )  
• list[5] = 44



7)  $n=7$   $i=7$   
index = 12  
 $j=6$   $\rightarrow$  si  
while ( $6 > 0$  y  $46 > 12$ )  
•  $list[7] = 46$   
 $j=5$   $\rightarrow$  si  
while ( $5 > 0$  y  $44 > 12$ )  
•  $list[6] = 44$   
 $j=4$   $\rightarrow$  si  
while ( $4 > 0$  y  $33 > 12$ )  
•  $list[5] = 33$   
 $j=3$   $\rightarrow$  si  
while ( $3 > 0$  y  $23 > 12$ )  
•  $list[4] = 23$   
 $j=2$   $\rightarrow$  si  
while ( $2 > 0$  y  $17 > 12$ )  
•  $list[3] = 17$   
 $j=1$   $\rightarrow$  no  
while ( $1 > 0$  y  $8 > 12$ )  
•  $list[2] = 12$   
return list