

Ordenamiento (Sorting):

Secuencia de valores de tipo A.

Es una funcion S de Naturales \rightarrow A tq dada $S(m) \rightarrow Am$.

Con los naturales en el rango entre 1 y n, siendo n la cantidad de elementos en la secuencia.

Entonces, la secuencia esta ordenada si para todo x de $1 \dots n$, $Ax < Ax+1$ ordenada en forma creciente.

Si para el mismo caso $Ax > Ax+1$, esta ordenada en forma decreciente.

Definicion de Ordenar: Dada una secuencia, es hacer una permutacion entre los naturales de forma tal que cumpla con la condicion ordenada.

Definicion general independiente del tipo de datos: lo que debe tener definido el tipo de dato es la operacion de comparacion ($<>$) como una relacion de orden total. Essto quiere decir que todos los elementos del dominio tienen que ser comparables entre si.

En general trabajamos con arreglos de enteros.

Vamos a analizar y comparar 6/7 algoritmos de ordenamiento.

Bubble sort:

Chequea si $An < An+1$, si lo es los cambia de lugar con un swap, luego realiza los mismo con $An+1 < An+2$ y sigue hasta llegar al final del arreglo, lista, etc. . .

~~~~~

Ej: 5 8 8 3 5 6 8 3 5 1 6 3 6 1 2 2 2 1 En este caso con 2 pasos basta para ordenar todo el arreglo. ~~~~~ Si tengo n datos, voy a necesitar n-1 iteraciones.

Especificacion recursiva del algoritmo:

`burb: Seq  $\rightarrow$  Seq(Arreglo), NumIt, Pos: Enteros`

`burbaux: Arr x NumIt x Pos  $\rightarrow$  Arr    a:Array; n,ni,p=Enteros`

`burb(a)=burbaux(a,n-1,1)                      n=cant. de datos`

`burbaux(a,ni,p)={    a                                              ni=0  
                     burbaux(a,ni-1,1)                              ni>0 y p=n  
                     burbaux(swap(a,p),ni,p+1)                      ni>0 y p<n`

`swapb: Arr x Ent  $\rightarrow$  Arr`

`swapb(a,pp)={    swap(a,pp,pp1)                      a[pp]>a[pp+1]  
                 a                                      a[pp]<=a[pp+1]`

swap: Arr x Ent x Ent -> Arr Es la funcion

---

## Seleccion:

Particiono la secuencia en 2 partes, el subconjunto de los datos ordenados y el subconjunto de los datos desordenados. Inicialmente los ordenados son una secuencia vacia.

Algoritmo iterativo: En paso (iteracion) selecciono el de mayor valor y lo paso al subconjunto de ordenados.

Ej:

```
5 8 8 8 8
3 3 6 6 6
8 5 5 5 5
1 1 1 3 3
6 6 3 1 2
2 2 2 2 1    n-1 iteraciones
```

Especificacion recursiva del algoritmo:

```
selaux(a,ni,p)={      a                                ni=1
                    selaux(swap(a,ni,k),ni-1,1,1)      ni>0 y p=n
                    selaux(a,ni,p+1,nuevok(a,p,k))      ni>0 y p<n
```

nuevok: Arr x Ent x Ent -> Ent

```
nuevok(a,p,k)= {    k      si a[p]<=a[k]
                   p      si a[p]>a[k]
```

---

## Insercion:

Primero ordeno los primeros 2 elementos, luego los primeros 3, luego los primeros 4, etc... Se llama algoritmo de insercion porque a medida que voy tomando un elemento mas, lo inserto en el lugar correcto.

Ej:

```
5 8 8 8 8
3 5 5 6 6
8 3 3 5 5
1 1 1 3 3
6 6 6 1 2
2 2 2 2 1    n-1 iteraciones
```

Ventajoso en una lista ya que requiere menor cantidad de procesos para agregar elementos en el medio o al final.