Rafael Enrique Almaraz Martínez 01/09/2017

Reporte: Algoritmos de ordenamiento

Bubble: Este algoritmo de ordenamiento realiza un cierto número de vueltas al arreglo de números comparándolos por parejas acorde a como sea requerido, de menor a mayor o de mayor a menor, sin embargo es un tanto ineficiente porque terminaría realizando un numero exagerado de operaciones.

```
Prodecimiento DeLaBurbuja (a0,a1,a2,…a(n-1))
        Para i<2 hasta n hacer
                Para j<0 hasta n-i hacer
                        Si aj >aj+1 entonces
                                aux<aj
                                aj<aj+1
                                aj+1<aux
                        fin si
                fin para
        fin para
fin procedimiento
```

Insertion: A diferencia de bubble, insertion en una sola vuelta acomoda los términos del arreglo ya que las comparaciones las ejecuta en conjuntos cada vez mayores empezando por parejas hasta llegar al tamaño total del arreglo.

1. Llama insert para insertar el elemento que comienza en el índice 1 en el índice 0 del subarreglo ordenado.
2. Llama insert para insertar el elemento que comienza en el índice 2 en los índices del 0 al 1 del subarreglo ordenado .
3. Llama insert para insertar el elemento que comienza en el índice 3 en los índices del 0 al 2 del subarreglo ordenado .
4. Por último, llama insert para insertar el elemento que comienza en el índice n-1*n*−1n, minus, 1 en los índices del 0 al n-2*n*−2n, minus, 2 del subarreglo ordenado.

Selection: El algoritmo de selección identifica el valor, ya sea menor o mayor, del arreglo y lo coloca en la primera posición que le corresponde, asi continuando con los demás elementos hasta que el listado este en el orden deseado.

```
para i=1 hasta n-1
   mínimo = i;
   para j=i+1 hasta n
      si lista[j] < lista[mínimo] entonces
         mínimo = j /* (!) */
      fin si
   fin para
   intercambiar(lista[i], lista[mínimo])
```

fin para

Quicksort: De manera más compleja pero más eficiente, quicksort escoge un valor cualquiera del arreglo para convertirlo en su pivote, una vez obtenido el pivote comparar el resto de los elementos del conjunto con este para situarlos a ambos lados, ya sean menores o mayores.

```
inicio
        variables A: arreglo[1..100] entero
        variables i,j,central:entero
        variables primero, ultimo: entero
        para i = 1 hasta 100
                        leer(A[i])
                Fin para
        primero = 1
        ultimo = 100
        qsort(A[],100)
Fin

Funcion qsort(primero, ultimo:entero)
                i = primero
                j = ultimo
                central = A[(primero,ultimo) div 2]
                repetir
                                mientras A[i]central
                                                j = j - 1
                                        fin mientras
                                si i < = j
                                                aux = A[i]
                                                A[j] = A[i]
                                                A[i] = aux
                                                i = i + 1
                                                j = j - 1
                                        fin si
                        hasta que i > j
                si primero < j
                                partir(primero,j)
                        fin si
                si i < ultimo
                                partir(i, ultimo)
                        fin si
        fin funcion qsort
```