

# Proyecto 3

## Algoritmos de Ordenación

Algoritmos y Estructuras de Datos II

7 de abril de 2006

Con la implementación del TAD arreglos de enteros (proyecto anterior) hacer 2 algoritmos que ordenen instancias del mismo. Los algoritmos se deben elegir de los vistos en clase.

Para ello tener en cuenta:

1. Los programas en C que implementan el TAD arreglo del proyecto anterior deben ser exactamente los mismos.
2. Escribir los 2 algoritmos de ordenación en archivos separados (.c y .h).
3. Tener en cuenta al elegir los 2 algoritmos de los vistos en el teórico que la única manera de modificar al TAD arreglo es por medio de la función `arrInt_swap`.
4. Los algoritmos deben tomar un arreglo previamente llenado y devolver el mismo pero ordenado. No deben devolver uno nuevo. Decidir cual es la signatura más conveniente para ello.
5. Hacer en otro archivo separado el bloque `main` con las mismas opciones que el del proyecto anterior, agregando una opción que permita al usuario ordenar el **mismo** arreglo por ambos métodos. Al terminar la ordenación se debe mostrar por pantalla el resultado de ambos métodos y el tiempo que tardo cada algoritmo en realizarla.

**Ayuda:** Para las funciones que manipulan el tiempo en C ver la info page `libc->Date and Time`.

Para hacer todos los programas seguir la siguientes directivas generales:

- Ocultar toda la información posible sobre la implementación del TAD al escribir los archivos con los algoritmos de ordenación y con el bloque `main`.
- Poner los bloques `ifndef` en cada archivo .h.
- En el (o los) .h con los algoritmos de ordenación especificar con `pre` y `pos` cada prototipo de los procedimientos expresando solo propiedades abstractas sobre arreglos. Agregar también descripción y uso (solo en los .h).
- Hacer lo mismo en el (o los) .c pero con las propiedades sobre el algoritmo concreto (o sea agregar invariantes).
- Demostrar la corrección de los algoritmos.
- Incluir con la directiva `#include` la mínima cantidad de headers que le harán falta a los algoritmos para poder ordenar el TAD arreglo en el (o los) .c donde se implementan los algoritmos de ordenación. Hacer lo mismo en el .c del bloque `main`.

- Hacer `Makefile` compilando todos los archivos con las opciones `-ansi`, `-pedantic` y `-Wall`. Ver que no haya mensajes de error o advertencias.
- No usar ninguna variable global.
- No dejar “memory leaks”.