



Algoritmos y Estructuras de Datos II

Trabajo Práctico 6 Búsqueda y Ordenamiento

OBJETIVOS

- Conocer los conceptos de búsqueda, clasificación(ordenamiento) e intercalación.
- Aprender a utilizar los métodos de búsqueda (interna, externa) según donde los elementos estén almacenados.
- Aprender a implementar soluciones para la búsqueda de máximos y mínimos.
- Diferenciar los distintos métodos de búsqueda y aplicarlos según sus ventajas y desventajas.
- Aprender a analizar la complejidad de los distintos métodos.
- Aprender a implementar soluciones para el ordenamiento de arreglos tipificado de acuerdo a su categorización.

COMPETENCIAS

- Identificar, formular y resolver problemas mediante programación.
- Utilizar de manera efectiva técnicas y herramientas de aplicación para desarrollar software.
- Desempeñarse de manera efectiva en equipos de trabajo.
- Aprender en forma continua, autónoma y de manera colaborativa.

METODOLOGÍA

- El alumno deberá resolver individualmente los ejercicios propuestos.
- El alumno deberá codificar las soluciones en el lenguaje de programación C.
- Realizar consultas a través del canal de slack correspondiente a su comisión ó del aula virtual de la asignatura.

DURACIÓN

De acuerdo a la planificación de la asignatura, se deberán utilizar para la resolución de los ejercicios de esta serie, dos clases prácticas.

EJERCICIOS DE BÚSQUEDA

1. Escribir un programa que permita determinar si un número entero ingresado por teclado, **existe en un vector**. Para ello será necesario, disponer de lo siguiente:
 - a) Una función que permita cargar elementos en un vector de números enteros.
 - b) Una función que permita visualizar los elementos almacenados en el vector.
 - c) Una función que devuelva verdadero en el caso que el elemento a buscar exista en el vector, y falso en caso contrario.
2. Escribir una función que permita encontrar un valor en un arreglo de números reales, según el método de **búsqueda binaria**. La función debe devolver *true* en el caso que el elemento a buscar exista dentro del arreglo y *false* en caso contrario. Probar la función en el bloque principal de un programa C.
3. Escribir un programa que permita ingresar por teclado la cantidad de elementos de un vector, y cada uno de los valores de los elementos del vector. Además, escribir funciones para obtener:
 - a) el **máximo** valor almacenado en el vector
 - b) el **mínimo** valor almacenado en el vector
 - c) la media de todos los valores
 - d) la semisuma (promedio entre el máximo y mínimo).
4. Dada una matriz A de 5 filas y 4 columnas, determinar el valor máximo y mínimo de la matriz, indicando además las posiciones en las que se encuentran estos valores.
5. En un concurso de baile en el que hay 10 participantes que bailan 7 estilos diferentes, se almacenan las puntuaciones (números reales con 1 dígito para la parte decimal ^(*)) en una matriz de 10 filas por 7 columnas. La puntuación final de cada participante se obtiene como la suma de sus puntuaciones excluyendo la mayor y la menor (con lo cual se suman 5 puntuaciones para cada participante). Diseñar un programa que indique quien es el ganador del concurso.
 (*) La matriz debe ser cargada con cualquier expresión matemática que genere número reales correspondientes a la puntuación.

EJERCICIOS DE ORDENAMIENTO

1. Escribir un programa que permita ingresar 10 valores reales por teclado. Luego escribir funciones que permitan:
 - a) Ordenar un vector de menor a mayor por el método **directo de burbuja**.
 - b) Ordenar un vector de menor a mayor por el método **directo de selección**.
 - c) Ordenar un vector de menor a mayor por el método directo de inserción.
2. Desarrollar un programa que permita ingresar 10 valores enteros en cualquier orden en un vector. Luego:
 - a) Escribir una función para ordenar los elementos por algún método de ordenación.
 - b) Escribir una función para mostrar los elementos ordenados.
 - c) Escribir una función que reciba un valor a buscar y un vector en donde buscar el ese elemento, y realice la búsqueda a través del método de **búsqueda binaria**. En el caso que el elemento exista en el vector, se debe retornar la posición dentro del vector, o -1 en caso contrario.
 - d) En el programa principal, se deben utilizar las funciones y mostrar por pantalla el resultado de la búsqueda, indicando si el elemento ha sido encontrado o no.

3. Escribir un programa que permita generar un arreglo de 10 valores enteros que se generen de forma aleatoria. Luego:
 - a) Escribir una función para ordenar de menor a mayor los elementos del vector por el método logarítmico **Shell**.
 - b) Escribir una función para ordenar de mayor a menor los elementos del vector por el método logarítmico **QuickSort**.