



Algoritmos y Estructuras de Datos II

Práctico Nro 9 - Parte 1: Árboles

OBJETIVOS:

Que el alumno:

- Se familiarice con los conceptos de Árboles y Grafos.
- Aprenda a implementar las distintas formas de este tipo de estructuras (árboles y grafos).
- Implemente soluciones recursivas para el tratamiento de estructuras de datos no lineales (árboles y grafos).
- Continúe en el desarrollo de implementar las soluciones de problemas con un enfoque estructurado.

METODOLOGÍA

- Lectura de la conceptualización de recursividad.
- Lectura de la conceptualización de estructuras de datos no lineales: Árboles y Grafos.
- El alumno deberá resolver individualmente los ejercicios propuestos
- Se podrá realizar trabajos en grupos para consolidar conceptos, comprensión de lo solicitado y alternativas de solución.
- El alumno deberá codificar las soluciones que proponga de cada uno de los ejercicios propuestos en las clases prácticas de laboratorio.
- Interactuar en el aula virtual de la asignatura.

DURACIÓN

Según planificación de la asignatura se deberán utilizar para la resolución de los ejercicios de esta serie, no más de dos (2) clases prácticas.

EJERCICIOS PROPUESTOS

1. Construir un árbol binario correspondiente a una lista de números.
 - a) Nodo raíz del árbol: primer elemento de la lista
 - b) Los siguientes nodos serán descendientes derechos si son mayores y descendientes izquierdo si son menores. (Tener en cuenta el orden de la lista dada).
 - c) Verificar con los siguientes valores: **4 19 -7 49 100 0 22 12**
2. Recorrido de un árbol. Indique el recorrido en pre-orden, in- orden y post-orden del árbol del ejercicio anterior (Ej. 1). Mostrar el recorrido por pantalla.

3. Diseñe un algoritmo que cree un ÁRBOL BINARIO, el contenido de los nodos debe ser de tipo entero, y que posea los procedimientos para insertar, devolver el número de nodos y recorrer el árbol.
 - a) El procedimiento *Insertar* debe emitir una leyenda '*Es un hijo izquierdo*' o '*Es un hijo derecho*', según corresponda.
 - b) Escribir la función que devuelva el número de nodos del árbol binario.
 - c) El procedimiento Recorrer debe mostrar el contenido del nodo.
4. Construir un árbol binario de búsqueda, donde el contenido de cada nodo sea un entero positivo, y que permita las siguientes acciones:
 - a) Insertar nodos en el árbol.
 - b) Saber si el árbol está vacío.
 - c) Buscar un elemento en el árbol.
 - d) Eliminar un nodo del árbol.
5. Construir un árbol binario de búsqueda, donde el contenido de cada nodo sea un valor de tipo **caracter**, y que permita las siguientes acciones:
 - a) Insertar nodos en el árbol.
 - b) Eliminar un nodo del árbol.
 - c) Saber si el árbol está vacío.
 - d) Una función que implemente la búsqueda binaria de un elemento del árbol y que devuelva **true** en caso de encontrarlo y **false** en caso contrario.
 - e) Una función para concatenar los elementos del árbol mediante el acceso en **InOrden**.