

GUÍA DE TRABAJOS PRÁCTICOS

CURSADA 2023 – 1ER CUATRIMESTRE

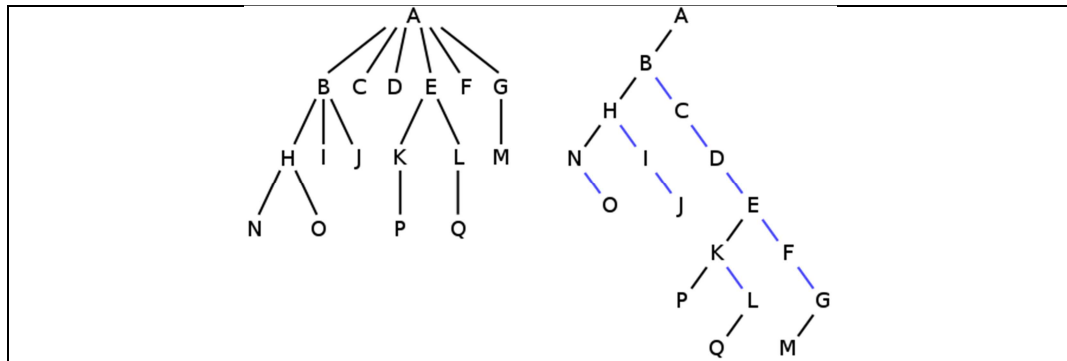
Práctica VI: Árboles

En cada caso se debe implementar el TAD descrito y una aplicación que, utilizando el tipo abstracto de dato, permita al usuario el ingreso de una o más instancias (según sea necesario), la aplicación de cualquiera de las operaciones y luego la obtención del resultado.

1. Construir el TAD de árboles correspondiente a la implementación de “Punteros” teniendo en cuenta:
 - a. Definir la estructura de datos en función a las especificaciones dadas en clase y documentadas en los PDF que se encuentran en el aula virtual de la asignatura.
 - b. Se las debe implementar y respetar estrictamente los nombres y parámetros de cada una de las operaciones del TAD especificadas.
 - c. Se las debe probar y testear de forma tal que se pueda asegurar que la TAD funciona correctamente.
2. Dado un **árbol binario** no vacío determinar:
 - a. Retornar una lista con todos los nodos terminales u hojas.
 - b. Retornar en una estructura todos los nodos interiores (los que no son ni hojas ni raíz)
 - c. Buscar todas las ocurrencias de una clave dentro del árbol. Retornar la posición de cada ocurrencia (puntero al nodo).
 - d. Determinar la complejidad algorítmica de los puntos “a”, “b”, “c”.
3. Para un **nodo** del árbol binario determinado:
 - a. Indicar el nombre del nodo padre (clave).
 - b. Listar los hijos (solo las clave/s).
 - c. Listar el hermano (solo la clave).
 - d. Calcular el nivel en el que se encuentra.
 - e. Calcular la altura de su rama (Altura del Subárbol)
 - f. Listar todos los nodos que están en el mismo nivel (solo la clave).
4. Para Árboles “N-arios”:

Un árbol n-ario puede ser representado como binario utilizando la *transformación de Knuth*. Esto puede ser útil para manejar árboles n-arios en estructuras de almacenamiento fijo, sin necesidad de conocer el “n” del árbol. Se propone un árbol binario derivado del n-ario, tal que para cada nodo del árbol n-ario, su primer hijo es el hijo izquierdo en el árbol binario, y los hermanos de cada nodo son sus hijos derechos.

Ejemplo: (n-ario a la izquierda, representado en binario a la derecha)



- Dado un árbol n-ario, devolver en forma de lista el resultado del recorrido en anchura del árbol (solo las claves).
 - Escribir una función que, dado un árbol n-ario, devuelva el número (cantidad) de hojas de dicho árbol.
 - Escribir una función booleana que dados dos árboles generales determine si tienen la misma estructura, sin importar los datos del mismo (árbol similar).
 - Retornar el padre de un nodo del árbol (tipo_elemento).
 - Retornar los hermanos de un nodo del árbol (lista de hermanos).
- Para Árboles Binarios de Búsqueda Balanceados “AVL” (Según el TAD)
 - Mostrar el resultado de insertar 5, 3, 8, 15, 9, 13, 6, 11, 10 y 12 en un árbol AVL inicialmente vacío (indicando las rotaciones realizadas).
 - Dibujar el árbol AVL que resulta de realizar las siguientes inserciones: 1, 11, 10, 12, 2, 8, 4, 1, 3. A continuación, eliminar los elementos: 18, 14, 17.
 - Realizar las operaciones que se detallan, marcando para cada nodo su factor de equilibrio en cada momento. En caso de producirse desequilibrio, indicar la causa y explicar con detalle qué operación se ha utilizado para resolverlo. Inserciones: 5, 11, 2, 19, 9, 10, 14, 6 y 15. Eliminaciones: 14, 9 y 10.
 - Para Árboles “B” y “B+” (Según el TAD)
 - Dibujar el árbol B de orden 5, resultado de insertar los siguientes datos: 13, 16, 3, 12, 15, 8, 4, 19, 5, 9, 21, 11.
 - Dibujar el árbol B de orden 5 que resulta a partir de la siguiente entrada de datos: A, D, M, P, Z, B, J, N, G, U, R, K, E, H, O, L, F, S, C, T.
 - Dibujar el árbol B de orden 5 que resulta de realizar las siguientes inserciones: 1, 7, 16, 12, 11, 14, 18, 13, 10, 15, 21, 19, 6, 4, 3, 2, 4, 22 y 28. A continuación, eliminar los elementos: 19, 10 y 14.
 - Determinar si dos árboles binarios son equivalentes.
 - Dado un árbol “n-ario” se pide:
 - Determinar la altura del mismo.
 - Determinar el nivel de un nodo.
 - Listar todos los nodos internos (solo las claves).
 - Determinar si todas las hojas están al mismo nivel.

9. Generar un algoritmo, recursivo o no, que permita construir un árbol binario de búsqueda balanceado (AVL) a partir de un árbol binario sin un orden determinado. Comparar las alturas de ambos árboles. Determinar la complejidad algorítmica.
10. Dada una serie de números generados al azar, cargar la misma serie en un árbol binario de búsqueda y en un árbol binario balanceado "AVL". Comparar la altura de ambos árboles. Repetir el proceso "n" veces. ¿Qué puede concluir al respecto?

IMPORTANTE !!!

- Los prácticos podrán ser resueltos en grupos de **3 (tres) a 6 (seis)** alumnos. Condición indispensable para que los TPs sean evaluados.
- Todos los prácticos son obligatorios.
- Los prácticos deben ser entregados en tiempo y forma en las fechas que los docentes estipulen (serán determinadas según la cursada y avisadas mediante la plataforma virtual).
- Los prácticos NO se recuperan por lo que si son entregados en condiciones incorrectas de funcionamiento, fuera de término o fuera de formalidad (abajo descripto) se consideran desaprobados.
- Se deben presentar todos los ejercicios del tp. Se deben tener como mínimo un 60% de los ejercicios correctos para aprobar la práctica, con 2 de los 3 últimos ejercicios funcionando correctamente. Se evalúan tanto los errores de lógica como los de ingreso de datos. Deben controlar los ingresos de datos, tales como blancos, ingreso de números correctos, etc.
- Los prácticos deben presentarse con la formalidad:
 - Carátula Impresa (Fecha de Entrega, Integrantes de Grupo, Título del Práctico a Entregar).
 - Se tomará asistencia en el momento de entrega del TP.
 - El código fuente debe estar disponible en GIT HUB, con acceso a los docentes.

El lenguaje de programación será "C". (Usar el foro para consultas sobre el TP)

FECHA DE ENTREGA: 31/05/2023 (17 hs) - (Luego se confirmará cómo será la presentación).

Profesor a Cargo de la Asignatura: Mario Perello.

Trabajos Prácticos a Cargo de: Claudia Reinaudi, José Racker, Pablo Chale y Mariano Goldman.