

Nombre y apellido: _____

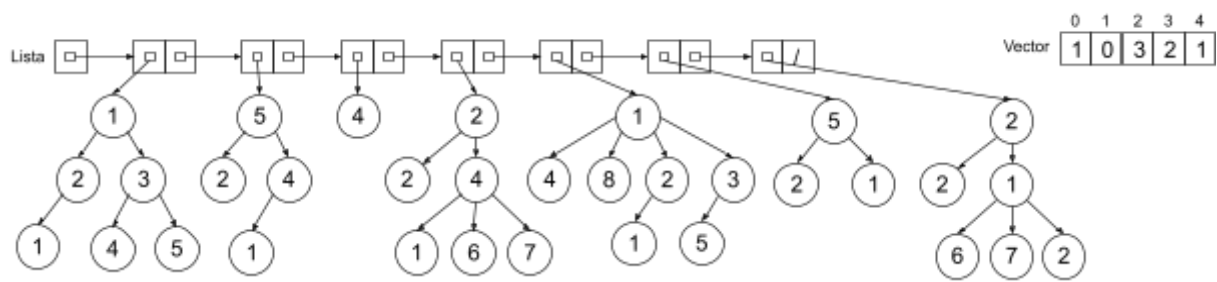
Ejercicios (puntaje)			Nota
1 (3 pts)	2 (3.5 pts)	3 (3.5 pts)	

CONSIDERACIONES: RESOLVER CADA EJERCICIO EN UNA HOJA DIFERENTE. LOS EJERCICIOS QUE NO ESTÉN CORRECTOS EN UN 50% NO SUMARÁN PUNTOS PARA LA NOTA FINAL. PARA CADA EJERCICIO DEBE **DEFINIR EL TIPO DE DATO** DE CADA TDA UTILIZADO. **EN TODOS LOS CASOS QUE UTILICE ESTRUCTURAS QUE NO SEAN TDAs "ESTÁNDAR" DEBE DEFINIR LOS STRUCTS.**

Ejercicio 1: Crear una función (y las funciones auxiliares necesarias) para que **dado un Árbol binario (BT), sustituirlo por un árbol binario de búsqueda (BST)** con la misma información.

Encabezado función principal: void **bt_to_bst** (btn** root)

Ejercicio 2: Desarrollar funciones (principal y auxiliares) que, a partir de una lista dinámica enlazada de árboles n-arios (implementados con **TDA list** para la lista de hijos de cada nodo), devuelvan un **TDA vector** con la cantidad de árboles agrupados por su grado. Ejemplo:



Encabezado función principal: vector* **tree_count_degree** (ntn* root, int n);

Ejercicio 3: Desarrollar las funciones necesarias para **determinar si G' es un grafo equivalente a G**. La función principal debe devolver 1 si son iguales o 0 si no lo son.

Consideraciones:

- G' (**g1**) está implementado con una **matriz de adyacencia** que representa un grafo dirigido (digrafo).
- G (**g2**) está implementado con **listas de adyacencia** que representan un grafo dirigido (digrafo).
- Ambos grafos poseen aristas ponderadas.
- Ambos grafos son conexos.
- Asumir que los vértices de ambos grafos coinciden en los índices.

Un digrafo G' equivale a G si:

1. G' y G tienen la misma cantidad de vértices.
2. Las aristas de G' están presentes en G con el mismo peso y dirección y viceversa.

Encabezado función principal: int **same**(graph_matrix* g1, graph_list* g2);