

Práctico 8

Estructuras de datos no lineales - Árboles

NOTA: Los ejercicios deberán entregarse completos en un archivo llamado `arbol.h`, siguiendo los criterios aconsejados por la práctica y con los controles adecuados, etc.

EJERCICIO 1

Programar una función que liste los nodos de un árbol en postorden.

a) Implementar como una función de aplicación con el siguiente prototipo:

```
template<class T>
void listarPostOrden(ArbolBin<T> &);
```

b) Implementar como una operación básica de árboles con el siguiente prototipo:

```
void ArbolBin<T>::listarPostOrden();
```

EJERCICIO 2 (Obligatorio)

Programar una función de aplicación que cuente los nodos hoja de un árbol (`ArbolBin<T>`).

```
template<class T>
size_t contarNodosHojas(ArbolBin<T> &);
```

EJERCICIO 3

Programar una función que permita cargar un ABB ([árbol binario de búsqueda](#)).

A. Implementar como:

- función de aplicación y
- operación básica de árboles.

B. Programar una función de testeo que cargue el árbol a partir de valores enteros (`int`) generados aleatoriamente en el rango $[-N, N]$.

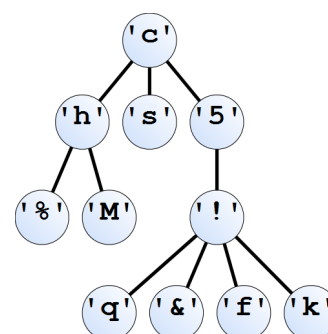
```
/**
 * @brief Prueba La carga en un ABB con valores enteros entre [-N, N]
 * @param cant Cantidad máxima de números a insertar en el ABB
 * @param N Rango de valores
 */
void test_abb(unsigned int cant, unsigned int N);
```

Nota: [Simulador](#) para visualizar la carga de un ABB.

EJERCICIO 4

Con el siguiente árbol n-ario se pide:

A. Grafique la representación del mismo como una lista de adyacencia.



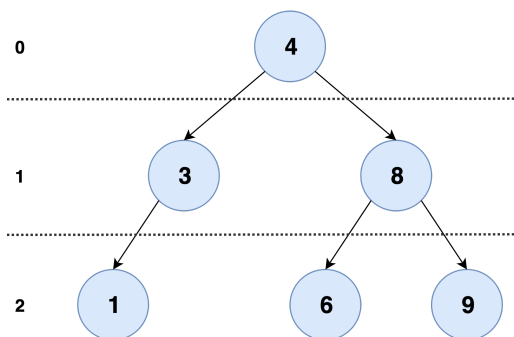
- B. Grafique la representación del mismo como una matriz de adyacencia.
- C. Transformar en un árbol binario utilizando la transformada de Knuth.
 - a. Grafique el árbol resultante como un array unidimensional.

EJERCICIO 5 (Obligatorio)

Programar una función de aplicación que liste los nodos de un árbol por niveles respetando el siguiente prototipo:

```
list<T> listarPorNivel(ArbolBin<T> &);
```

Para su implementación puede llamar a otras funciones y/o utilizar el TDA FilaA o Queue de STL.



[4, 3, 8, 1, 6, 9]