

Estructuras de Datos no Lineales

Práctica 4

Problemas de árboles generales

PASOS A SEGUIR

1. Implementar el TAD *árbol general* con las dos representaciones estudiadas, vector de listas de hijos y enlazada.
2. Escribir módulos que contengan las implementaciones de los subprogramas demandados en cada problema.
3. Para cada uno de los problemas escribir un programa de prueba, independiente de la representación del TAD elegida, donde se realicen las llamadas a los subprogramas, comprobando el resultado de salida para una batería suficientemente amplia de casos de prueba.

PROBLEMAS

1. Implementa una función genérica para leer desde la entrada estándar y en preorden los elementos de un árbol. Esta función *template <typename T> void leerAgen(Agen<T>& A, const T& fe)* construirá el árbol A con la estructura y elementos leídos de la entrada estándar. El segundo parámetro será un elemento que se utilizará para indicar el final de la entrada pero no se debe incluir en el árbol.

Pista: Implementa una función auxiliar para leer desde la entrada estándar y en preorden los descendientes de un nodo *n* de un árbol A:

template <typename T> void leerDescendientes(typename Agen<T>::nodo r, Agen<T>& A, const T& fe);

2. Implementa un subprograma, que dado un árbol general, nos calcule su grado.
3. Implementa un subprograma que, dados un árbol y un nodo dentro de dicho árbol, determine la profundidad de éste nodo en el árbol.
4. Se define el desequilibrio de un árbol general como la máxima diferencia entre las alturas de los subárboles más bajo y más alto de cada nivel. Implementa un subprograma que calcule el grado de desequilibrio de un árbol general.
5. Dado un árbol general de enteros A y un entero x, implementa un subprograma que realice la poda de A a partir de x. Se asume que no hay elementos repetidos en A.