

Estructuras de Datos

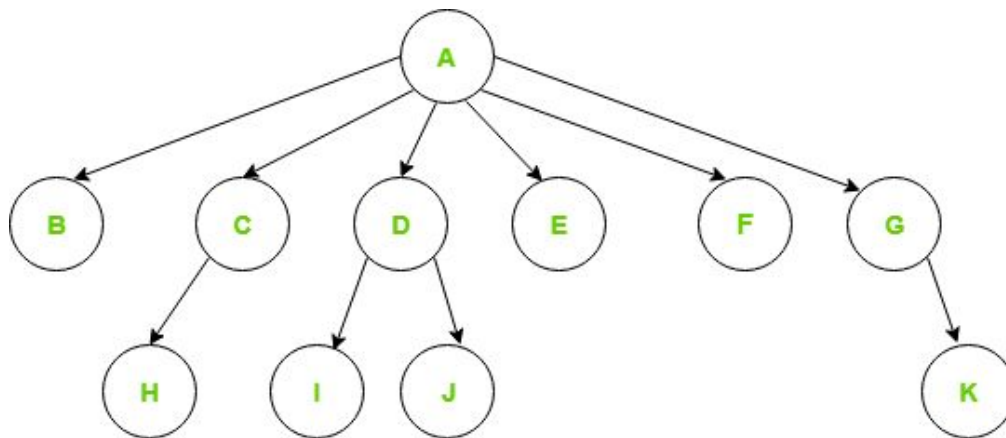
Árboles

Árboles Generales

En un **Árbol general** cada nodo puede poseer un número indeterminado de hijos. Esto nos genera ciertas complicaciones a la hora de pensar en su implementación. Vamos a ver dos enfoques para resolver este problema.

Árboles Generales

Nuestro primer enfoque tiene como idea el hecho de que al no saber cuántos hijos tiene cada nodo, se utilice una estructura dinámica para almacenarlos. Por ejemplo, podríamos pensar en una lista de hijos.



Árboles Generales

Es decir, tendríamos una implementación de este tipo. Con un dato genérico y usando nuestra declaración de listas genéricas, de la siguiente forma:

```
typedef struct _GTNode {  
    void* dato;  
    struct GList childs;  
} GTNode;  
  
typedef GTNode *GTree;
```

De esta manera, podemos implementar un **Árbol general**, que además, es genérico.

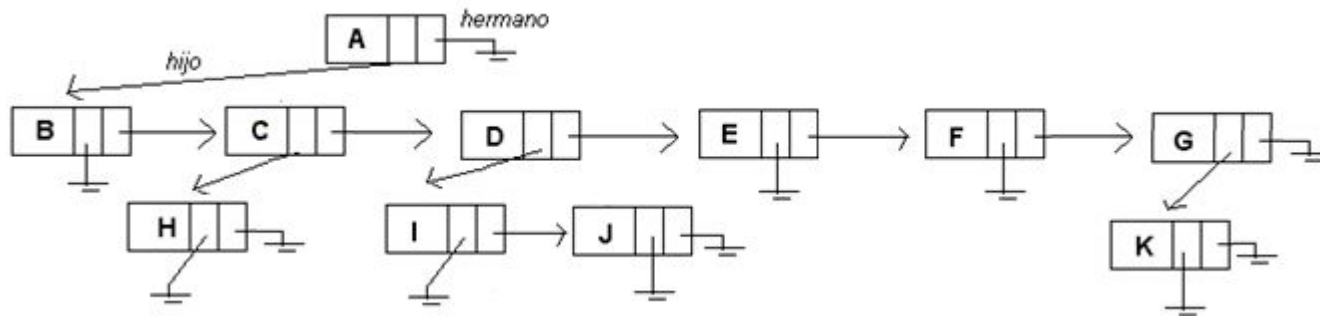
Nuestro segundo enfoque plantea la siguiente implementación de los nodos.

Como no se sabe de antemano cuántos hijos tiene un nodo en particular se utilizan dos referencias, una a su primer hijo y otra a su hermano más cercano.

La raíz del árbol necesariamente tiene la referencia a su hermano como *null*.

Árboles Generales

Es decir, tendríamos una implementación que generaría un árbol de esta manera, teniendo en cuenta el ejemplo del slide 3.



Árboles Generales

Notemos que, con este enfoque, todo árbol general puede representarse como un árbol binario donde el puntero a derecha apunta al hermano y, el puntero a izquierda apunta a su primer hijo.

En esta implementación, la raíz no tendría hermanos, por lo que sería siempre *null*.

Si se permite que la raíz del árbol tenga hermanos, lo que se conoce como *bosque*, entonces tendríamos que el conjunto de los bosques generales es isomorfo al conjunto de los árboles binarios.

En efecto, las propiedades vistas en los árboles binarios se siguen cumpliendo en los árboles generales.