

## **1.- ¿Por qué surgen los árboles?**

Porque se necesita una estructura de datos que no esté organizada de manera secuencial, sino jerárquicamente, para así poder plasmar situaciones no representables con otros tipos de estructuras lineales; además de para obtener búsquedas de órdenes logarítmicos.

## **2.- ¿Por qué la operación de Insertar y Eliminar son de $O(1)$ en la representación vectorial o de celdas enlazadas de árboles binarios?**

Para insertar, porque en la vectorial siempre añadimos en la última posición del vector que esté vacía y el acceso al vector es de orden  $O(1)$ . En la de celdas enlazadas, simplemente tenemos que apuntar el puntero que tiene el padre al nuevo nodo creado, lo cual también es de orden  $O(1)$ .

La operación de eliminar es de  $O(1)$  en la representación de celdas enlazadas porque la acción que lleva a cabo es la de liberar memoria y apuntar a  $\langle\langle\text{null}\rangle\rangle$  el puntero al nodo eliminado. En la representación vectorial, la operación eliminar es de  $O(1)$  porque el apuntador al nodo eliminado se queda en nodo nulo, y el nodo que estaba en la última posición mueve a donde estaba el nodo eliminado. Como en estas operaciones no tenemos que buscar nada, la operación se hace en coste constante.

## **3.- ¿Las operaciones insertar y eliminar del TAD Binario representado mediante un vector son de $O(1)$ ?**

Para insertar, porque en la vectorial siempre añadimos en la última posición del vector que esté vacía y el acceso al vector es de orden  $O(1)$ . En la de celdas enlazadas, simplemente tenemos que apuntar el puntero que tiene el padre al nuevo nodo creado, lo cual también es de orden  $O(1)$ .

La operación de eliminar es de  $O(1)$  en la representación de celdas enlazadas porque la acción que lleva a cabo es la de liberar memoria y apuntar a  $\langle\langle\text{null}\rangle\rangle$  el puntero al nodo eliminado. En la representación vectorial, la operación eliminar es de  $O(1)$  porque el apuntador al nodo eliminado se queda en nodo nulo, y el nodo que estaba en la última posición mueve a donde estaba el nodo eliminado. Como en estas operaciones no tenemos que buscar nada, la operación se hace en coste constante.

## **4.- ¿La eliminación de nodos de un árbol binario se puede conseguir con $O(1)$ , cuando se utiliza una representación vectorial con índice al padre, hijo izquierdo e hijo derecho?**

**Editada.**

En una representación vectorial, la eliminación de nodos es de orden constante.

## **5.- Ventaja de la representación de árboles binarios con celdas enlazadas frente a matrices.**

La principal ventaja es que la representación con celdas enlazadas es en memoria dinámica, mientras que con matrices es en pseudoestática. Con las celdas enlazadas ocupamos el espacio que necesitamos; en la de matrices ocupamos siempre el espacio máximo del árbol, el cual debemos establecer a priori para esta representación.

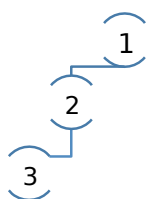
**6.- ¿Puede construirse de forma única un árbol binario dado, conociendo su preorden y el peso de cada nodo (número de nodos descendientes suyos)?**

No, para poder construir un árbol binario de forma unívoca, necesitamos conocer su inorden y su preorden o postorden. A partir del recorrido en preorden o postorden obtenemos el nodo raíz y, gracias al recorrido inorden sabemos que nodos pertenecen al subárbol izquierdo y quienes al subárbol derecho de dicho nodo y así recursivamente con los subárboles.

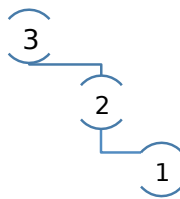
**7.- Sean A y B dos árboles binarios diferentes, indique si puede ocurrir simultáneamente que:  $\text{Pre}(A) = \text{Post}(B)$  y  $\text{Pre}(B) = \text{Post}(A)$ ? Editada.**

Si, cuando los dos árboles binarios están vacíos; cuando solo tienen la raíz; y cuando los árboles sigan esquemas similares al siguiente:

A:



B:



- $\text{Pre}(A) = 1, 2, 3$  ^  
 $\text{Post}(A) = 3, 2, 1$  ^  
 $\text{Pre}(B) = 3, 2, 1$  ^  
 $\text{Post}(B) = 1, 2, 3 \Rightarrow$   
 $\text{Pre}(A) = \text{Post}(B)$  ^  
 $\text{Pre}(B) = \text{Post}(A)$

**8.- ¿En qué situaciones es conveniente utilizar un vector de posiciones relativas? -> Examen 2015 Revisar**

Quando conozcamos de antemano el grado que tendrá nuestro árbol (número máximo de hijos), y además cuando este sea completo, de forma que optimicemos el espacio reservado en memoria para el árbol.

**9.- ¿A partir de dos recorridos cualesquiera conocidos, podemos conocer el árbol?**

No, para poder construir un árbol binario de forma unívoca, necesitamos conocer su inorden y su preorden o postorden. A partir del recorrido en preorden o postorden obtenemos el nodo raíz y, gracias al recorrido inorden sabemos que nodos pertenecen al subárbol izquierdo y quienes al subárbol derecho de dicho nodo y así recursivamente con los subárboles.

**10.- ¿Cuántos tipos de recorridos de árboles en anchura existen? Explica sus diferencias**

Solo existe un recorrido en anchura, que es el recorrido por niveles.

**11.- ¿Qué condición tienen que cumplir los elementos de un árbol para poder realizar las búsquedas con un coste menor que  $O(n)$ ?**

Deben tener una relación de orden y su disposición en el árbol debe ser la adecuada para que dicho árbol esté equilibrado.