

## Borrar un nodo de un árbol binario:

Supongamos un árbol binario con la siguiente estructura:

```
typedef struct
{
    int r,
    struct nodoArbol * i;
    struct nodoArbol * d;
}nodoArbol;

nodoArbol * A;
```

Sea la cabecera de la función borrar: *nodoArbol \* borrar (nodoArbol \* A, int dato)*

Para borrar un nodo hacemos esto:

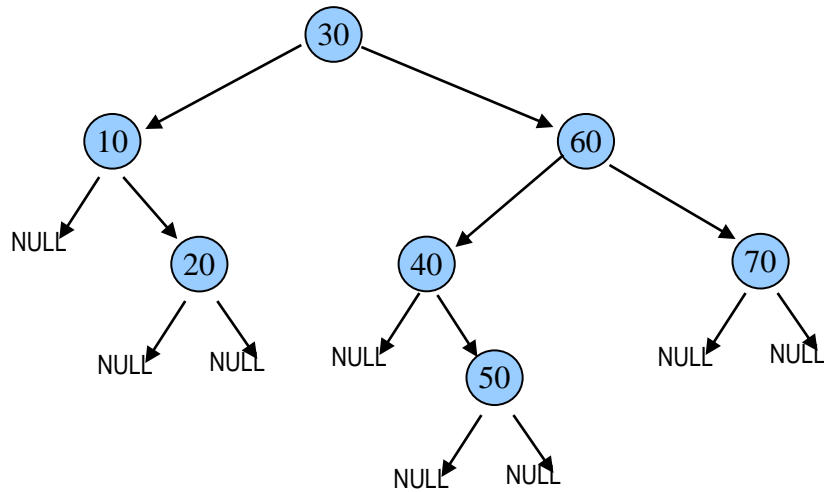
A = borrar (A, dato);

donde borrar(A, dato) es igual a:

Si A != NULL	Si dato == A->r	Si A->i != NULL	<b>nodoArbol * masDer = NMD(A-&gt;i);</b> A->r = masDer->r; A->i = borrar(A->i, <b>masDer-&gt;r</b> );	Al encontrar el nodo con el dato buscado, sobrescribo este dato con el dato del “nodo más derecho” del subárbol izquierdo (si existe nodo izquierdo)
		Sino si A->d != NULL	<b>nodoArbol * masIzq = NMI(A-&gt;d);</b> A->r = masIzq->r; A->d = borrar(A->d, <b>masIzq-&gt;r</b> );	Similar al caso anterior, pero considerando el otro subárbol.
		Sino (cuando A es hoja)	Free (A); A = NULL;	Cuando A es hoja, simplemente se libera la memoria, se la establece en NULL y se la retorna.
	Si dato > A->r		A->der = borrar (A->d, dato);	Se busca el dato recursivamente en el subarbol derecho
	Si dato < A->r		A->izq = borrar (A->i, dato);	Se busca el dato recursivamente en el subarbol izquierdo.
Si A == NULL				El elemento buscado no está en el árbol. No hacer nada, no hay else
		Al final de la función	return A	Siempre retornar A;

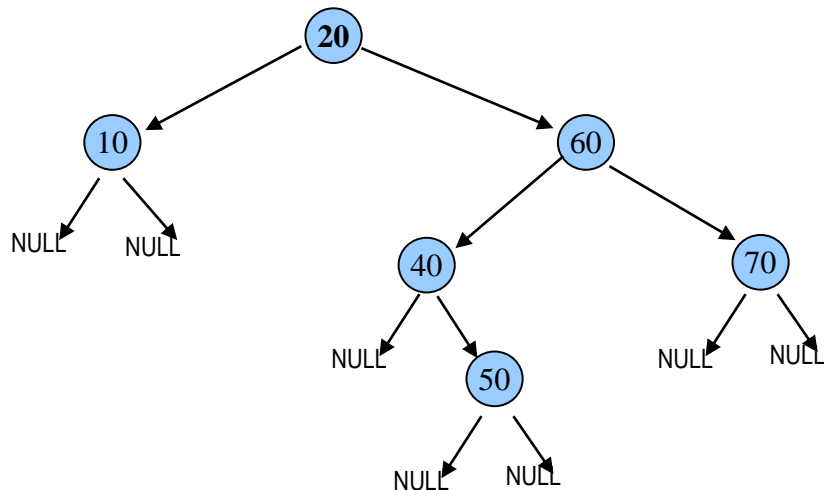
Al encontrar el elemento, si es hoja, simplemente se lo elimina; si no es hoja, debemos reemplazar el dato del nodo encontrado, con el dato siguiente en orden (el menor o el mayor). Este dato lo encontramos en el NMI(D) (nodo más izquierdo del subárbol derecho) o en el NMD(I) (nodo más derecho del subárbol izquierdo).

Pensar cómo se hacen estas dos funciones recursivas.



Por ejemplo, si se borra el nodo 30, el proceso será:

1. buscar el nodo 20 (nodo más derecho del subárbol izquierdo)
2. asignar al nodo 30 un dato (raíz) igual a 20.
3. asignar al hijo izquierdo del viejo nodo 30 (nuevo nodo 20) el resultado de borrar al subárbol izquierdo del viejo 30 el nodo con valor 20, o sea:
  - a.  $A \rightarrow \text{izq} = \text{borrar}(A \rightarrow \text{izq}, \text{NMD}(A \rightarrow \text{izq}))$ ;
  - b. Al ejecutar esta entrada en recursión, se ubica el nodo 20, que como es un nodo hoja (no tiene hijos), directamente se lo elimina del árbol.



alumno la codificación del algoritmo. Igualmente se puede preguntar en clase.