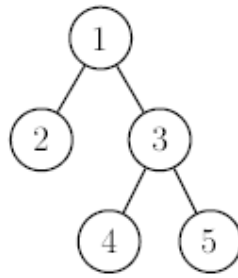


## Hojas profundas

En este control hay que implementar una función que, dado un árbol binario de enteros y un número entero no negativo  $k$ , determine el número de hojas cuya profundidad es mayor que  $k$ . Por ejemplo, para el siguiente árbol



la función devolvería 3 si  $k=0$  o  $k=1$ , devolvería 2 si  $k=2$  y devolvería 0 si  $k=3$ .

Aparte de implementar la función, hay que indicar razonadamente su complejidad (ten en cuenta que la implementación que usas del TAD Arbin es la eficiente –la que emplea la técnica de conteo de referencias– y que, por tanto, todas las operaciones del TAD tienen coste  $O(1)$  (asumiendo que el tipo de los elementos del árbol tenga operaciones de copia y destrucción de coste constante)).

---

### Detalles de implementación

Se proporciona el archivo `main.cpp` en el que se implementa la lógica de entrada / salida necesaria y se incluye un comentario sobre el formato de dicha entrada / salida así como varios ejemplos de salidas que el programa debería dar para los árboles correspondientes.

El código proporcionado no debe modificarse, salvo lo que corresponda a la implementación de la siguiente función

```
// Devuelve el número de hojas del árbol recibido cuya profundidad es mayor que un valor dado
// como parámetro.
// Parámetros:
//     a: árbol binario de enteros.
//     k: profundidad a superar para que una hoja sea contabilizada.
// Resultado:
//     N° de hojas de a con profundidad mayor que k.
unsigned int numero_hojas_mas_profundas_que(const Arbin<int>& a, unsigned int k);
```

así como la incorporación de todas aquellas funciones auxiliares que se consideren necesarias.

En este control, como en todos, hay que usar los TADs vistos en la asignatura y las implementaciones disponibles en el campus virtual. En particular, no se permite el uso de la STL.