

## Algoritmos y Estructuras de datos

### Ejercicios previos al tercer examen parcial

- Escribe una implementación de la función `cuenta_nodos`, la cual toma un apuntador al nodo raíz de un árbol binario, y devuelve cuántos nodos tiene el árbol. Las hojas (es decir, los nodos sin hijos) apuntan a `nullptr` como sus nodos hijos.

```
struct nodo {
    int valor;
    nodo* izq;
    nodo* der;
};

int cuenta_nodos(nodo* p) {
    if (p == nullptr) {
        return 0;
    } else {
        return 1 + cuenta_nodos(p->izq) + cuenta_nodos(p->der);
    }
}
```

- Escribe una implementación de la función `cuenta_apariciones`, la cual toma un apuntador al nodo raíz de un árbol binario además de un entero y devuelve cuántas veces aparece el entero en el árbol. Las inserciones en el árbol se realizaron como sigue: todos los valores del subárbol izquierdo de un nodo son menores o iguales al valor de dicho nodo, mientras que todos los valores del subárbol derecho son mayores. Las hojas (es decir, los nodos sin hijos) apuntan a `nullptr` como sus nodos hijos. Tu función no debe buscar en partes del árbol donde se sabe (por la regla de inserción) que el entero no aparece.

```
struct nodo {
    int valor;
    nodo* izq;
    nodo* der;
};

int cuenta_apariciones(nodo* p, int v) {
    if (p == nullptr) {
        return 0;
    } else if (v <= p->valor) {
        return cuenta_apariciones(p->izq, v) + (v == p->valor);
    } else if (v > p->valor) {
        return cuenta_apariciones(p->der, v);
    }
}
```