

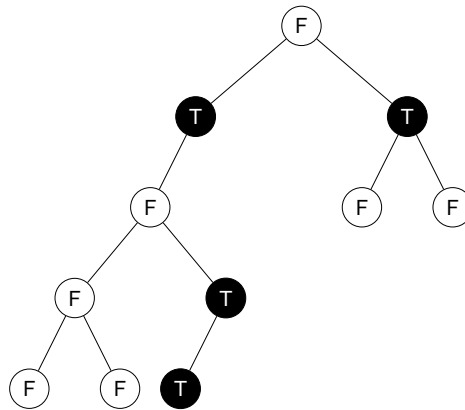
## 7.4 – Área más grande en un árbol binario

---

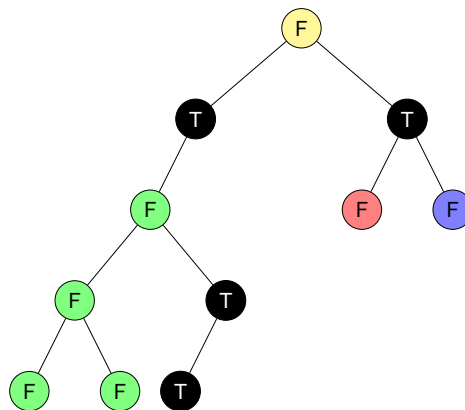
Estructuras de Datos  
Facultad de Informática - UCM

Estructuras de Datos  
Facultad de Informática - UCM

En este ejercicio consideramos árboles binarios de booleanos, en el que los nodos con valor `false` se consideran nodos vacíos, y los nodos con valor `true` se consideran *nodos barrera*. Por ejemplo, supongamos el árbol de la siguiente figura, en el que los nodos barrera se representan mediante T y los nodos vacíos se representan mediante F:



Los nodos barrera dividen el árbol binario en distintas *áreas*. Un área es un conjunto de nodos vacíos conectados entre sí por caminos que no pasan ninguna barrera. En el árbol anterior podemos distinguir cuatro áreas, señaladas con distintos colores en la siguiente figura. Dos nodos tienen el mismo color si y solo si pertenecen a la misma área:



Como puede verse, el área más grande es la formada por los nodos que se muestran en color verde. Este área tiene cuatro nodos, mientras que las restantes tienen un nodo cada una. El objetivo de este ejercicio es, dado un árbol de booleanos, determinar el tamaño (es decir, el número de nodos), del área más grande.

Define una función `area_mayor_sin_barreras` con la siguiente cabecera:

```
int area_mayor_sin_barreras(const BinTree<bool> &tree)
```

Esta función debe devolver el tamaño del área más grande del árbol de booleanos pasado como parámetro, suponiendo que los nodos barrera son aquellos que tienen el valor true. Por ejemplo, para el árbol mostrado en la figura anterior, la función debe devolver 4.

Indica el coste, en el caso peor, de la función anterior. El coste debe estar expresado en función del número de nodos del árbol de entrada.