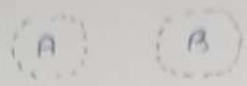
Planteamiento Práctica 2

Problema 1



similares

Los dos arboles y similares

entradas → las raices de los arboles

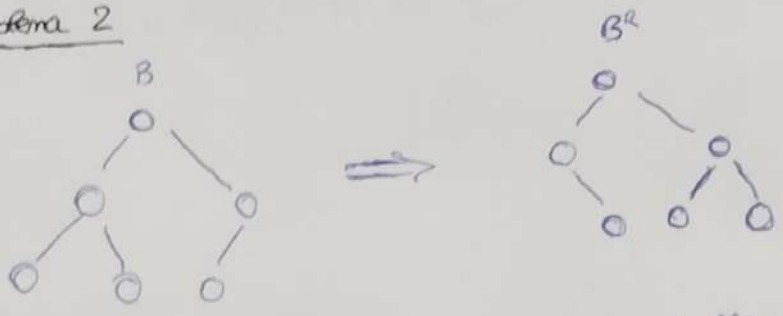
salida → booleano true si son similares, false caso contrario

Idea → empezamos con las raices del arbol A y B y compruebo sus hijos derecho y e hijo izquierdo. Si son iguales las llamadas recursivas poniendo como raiz el hijo derecho e izquierdo, para mirar subarboles

```
if ( A.hijoDrch(n) == B.hijoDrch(n) && A.hijoIzq(n) == B.hijoDrch(n) )
    return similares(A, A.hijoDrc(n), B, B.hijoDrech(n)) &&
    similares(A, A.hijoIzqd(n), B, B.hijoIzqd(n))

else
    return false
```

Problema 2



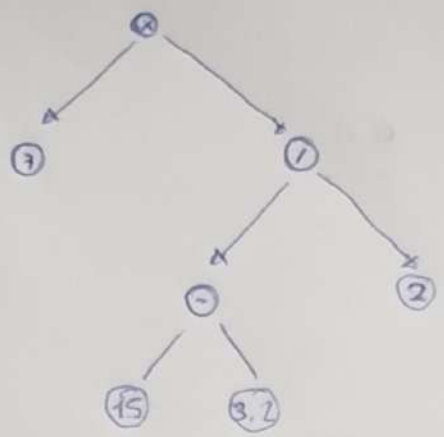
entrada → un árbol y otro vacío que se irá llenando

salida → arbol reflejado

Idea → crearas un árbol vacio nuevo y vamos recorriendo el árbol completo que recibimos y vamos poniendo el hijo derecho de B como hijo izquierdo del nuevo árbol y el hijo izquierdo de B como hijo derecho del nuevo árbol.

Recorrido en preorden

Problema 3



$7 \times [(15 - 3{,}2) / 2]$

a) Habrá que realizar un enum para los operadores

enum operador { suma = '+', resta = '-', producto = '*', division = '/' };

Para los operandos comprobar si es hoja el nodo ~~con la función correcta~~ comprobando si los hijos de ese nodo son nulos

b) Caso base: si es hoja devolveremos el valor

Caso general: switch comprobando cada caso de operador y realizar lo que proceda

calcular(A.hijoIzq(n), A) + calcular(A.hijoDrecho(n), A)

En caso de la suma por ejemplo