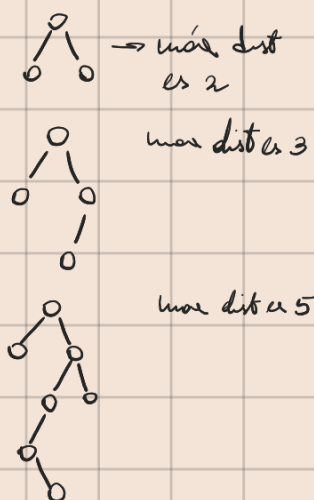
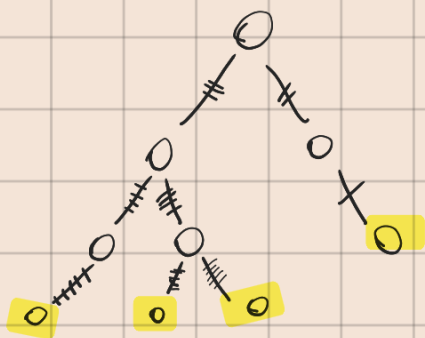
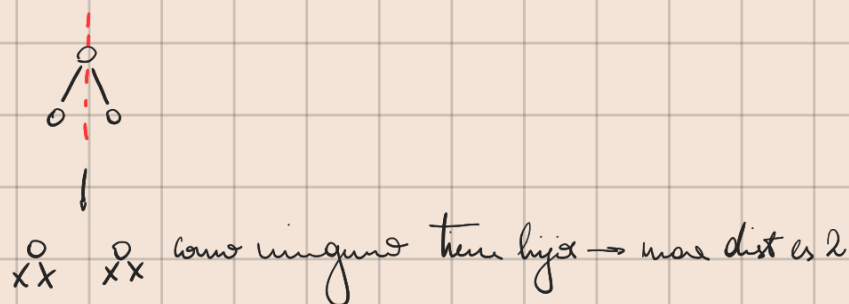


Ejercicio 6 (*DistanciaMáxima*) ★

Dado un árbol binario cualquiera, diseñar un algoritmo de dividir y conquistar que devuelva la máxima distancia entre dos nodos (es decir, máxima cantidad de ejes a atravesar). El algoritmo no debe hacer recorridos innecesarios sobre el árbol. **Hint:** para saber el camino más largo de un árbol, posiblemente necesite conocer más que sólo los caminos más largos de sus subárboles.



La única forma que tengo de dividir un árbol es entre hijo izquierdo o hijo derecho \rightarrow tengo que hacer el congreso en cada subárbol



Así puedo ir dividiendo. ¿Cuál va a ser mi con-
gruente?

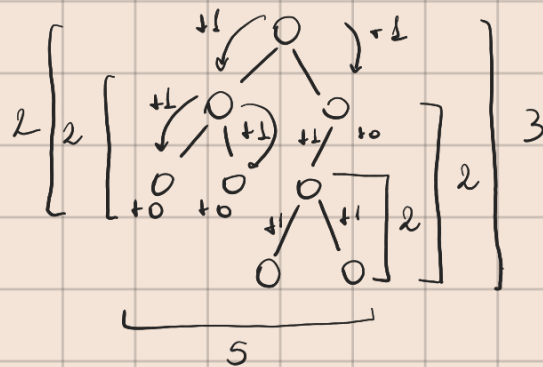
c Cuando llega a una hoja \rightarrow caso base es cuando es hoja el nodo en el que estoy



Buenos \rightarrow llega a una hoja c qué hago? Como en este caso la dist. usa de sumar ej: y una hoja no tiene pinto sumar 0

Divide \rightarrow divo al hijo derecho o al izquierdo o ambos según corresponda

Combine \rightarrow probablemente necesite elegir el máximo entre valores.



La distancia máxima entre 2 nodos siempre va a ser la cantidad de ejes que necesito atravesar para ir de la hoja que se encuentra más profunda en el subárbol derecho y la que más en el izquierdo. Si no hay subárbol derecho o izquierdo es la distancia de la hoja + profunda a la raíz.

DM (raiz, HI, HD)

if esHoja(raiz, HI, HD) v esNil(raiz, HI, HD) then
return 0

else

AI \leftarrow DM(HI, hijoIzq(HI), hijoDer(HI)) + 1

AD \leftarrow DM(HD, hijoIzq(HD), hijoDer(HD)) + 1

En resumen a ver que con 1 sola función NO sale. Hay una simplificación que podemos resolver con + facilidad \rightarrow Calcular máxima cantidad de ejes entre un subárbol izq y uno derecho
 \rightarrow la formulación que estamos haciendo apuntaba a esto

DM($raiz$, NI , ND)

if $esHija?(raiz) \vee esNil(raiz)$ then
return 0

else

$sr \leftarrow 2 + DM(NI, hijoIzq(NI), hijoDer(NI)) + DM(ND, hijoIzq(ND), hijoDer(ND))$

$SI \leftarrow DM(NI, hijoIzq(NI), hijoDer(NI))$

$SD \leftarrow DM(ND, hijoIzq(ND), hijoDer(ND))$

return $\max\{sr, SI, SD\}$

→ Compara el camino + largo del árbol y de sus subárboles