

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.

Decimos que B es árbol, implementado como lista de adyacencias.

$B \leftarrow$ nuestro árbol binario

$r \leftarrow$ raíz del árbol B

```
dfs(r):
    si r = Nil: //0(1)
        return (0,0)

    (izq_alt,izq_dist) = dfs(B[r][1])
    (der_alt,der_dist) = dfs(B[r][2])

    altura = 1 + máx(izq_alt,der_alt) //0(1)
    dist = máx(máx(izq_dist, der_dist), izq_alt+der_alt) //0(1)

    return(altura, dist)

DistanciaMáxima(B, r):
    (w,res) = dfs(r) //w no representa nada

    return res
```

Esto recorre recursivamente todos los nodos una única vez, es $\Theta(n)$.