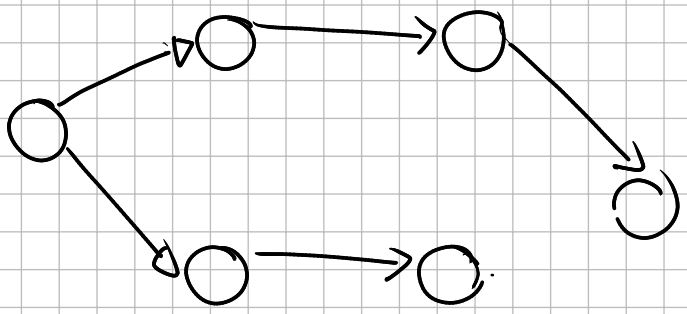
Arboles

Grafos aciclicos dirigidos.

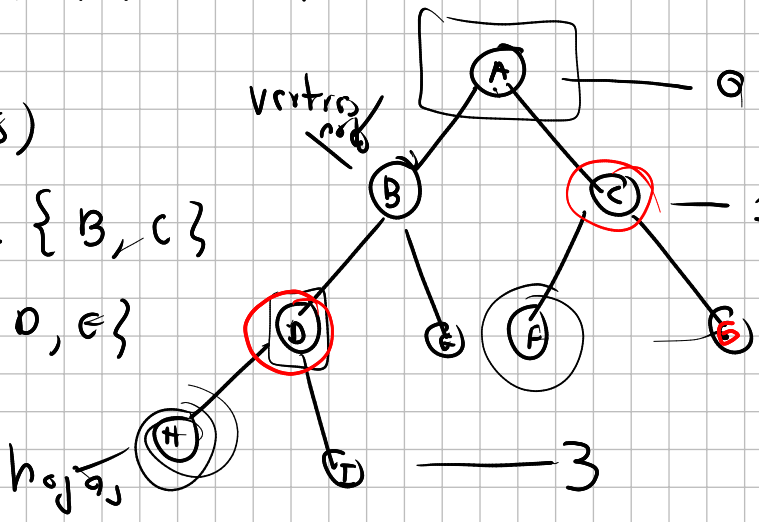


Raiz, Nodo interno

Hijos(s)

h(A) = {B, C}

h(B) = {D, E}



Padre / Madre

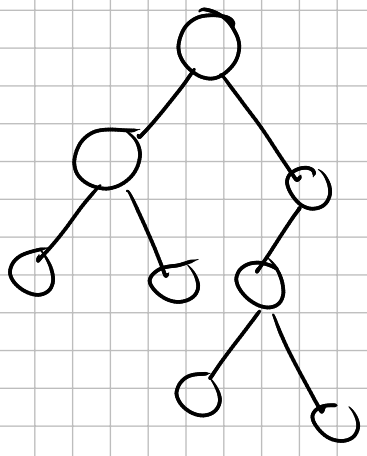
P(F) = C

P(C) = A

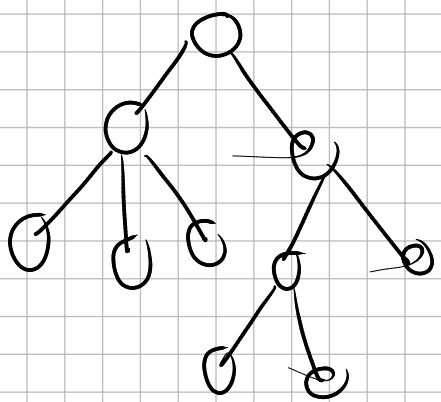
P(A) = null

Arboles m-ario (max m hijos cada nodo)

binario

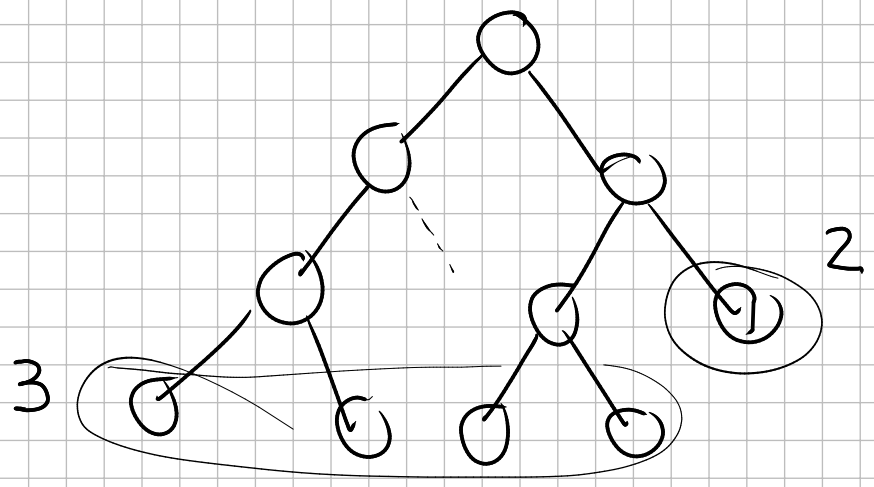


ternario / trinario

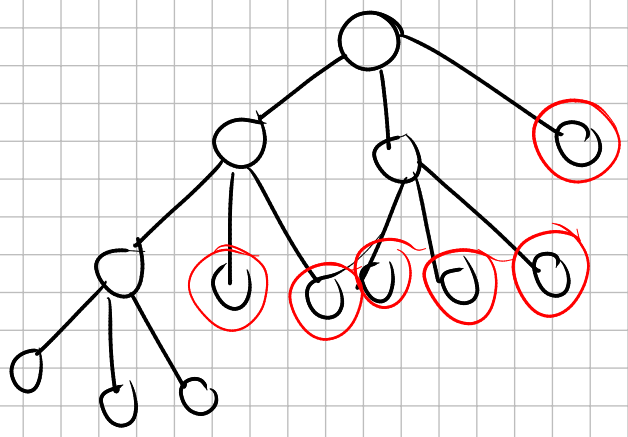


balanceado

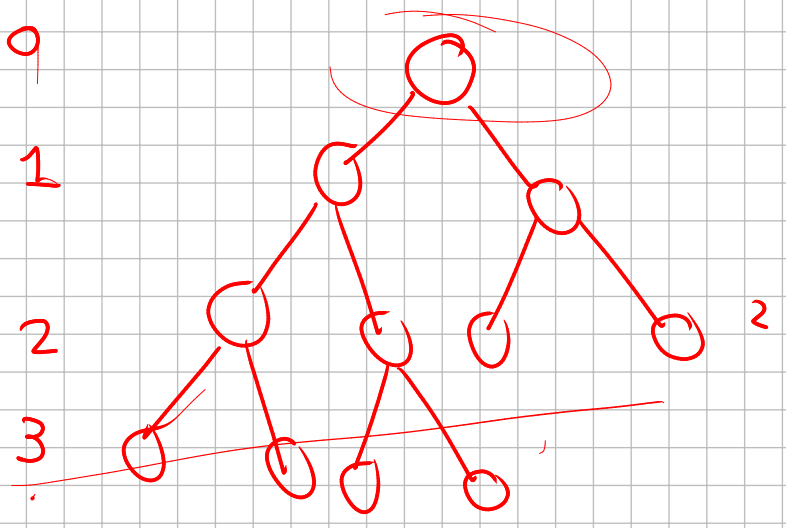
hojas: h o h - 1



Completo: todos nodos/vértice tienen exactamente m hijos



Arbol m-ario completo
Es un arbol completo y es balanceado



m^h

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$\max(2^3) = \max(8)$

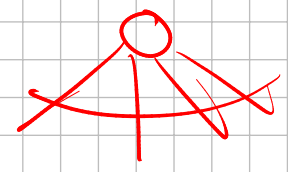
$\max(m^h)$ hojas

$m = 10$ $h = 15$

Vertices $h = 8$ 10^8

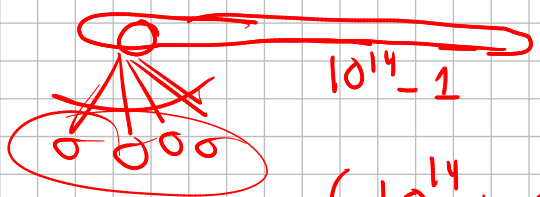
hojas $\max(10^{15})$

$\text{rango}(10^{14}, 10^{15})$



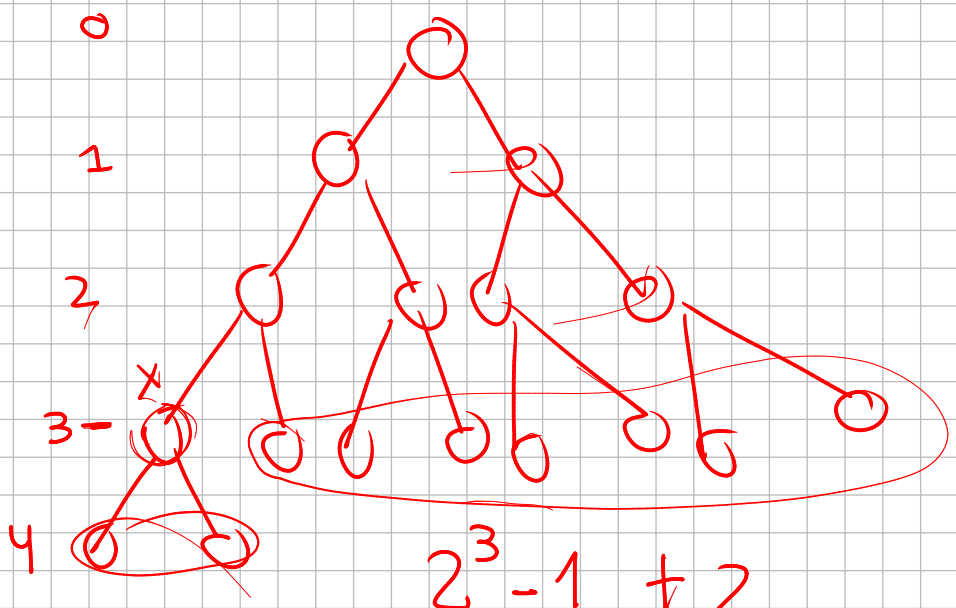
10^0
10^1
10^2
10^3
$\vdots$
10^{14}

Completo
m-ario



10

$(10^{14}+9, 10^{15})$



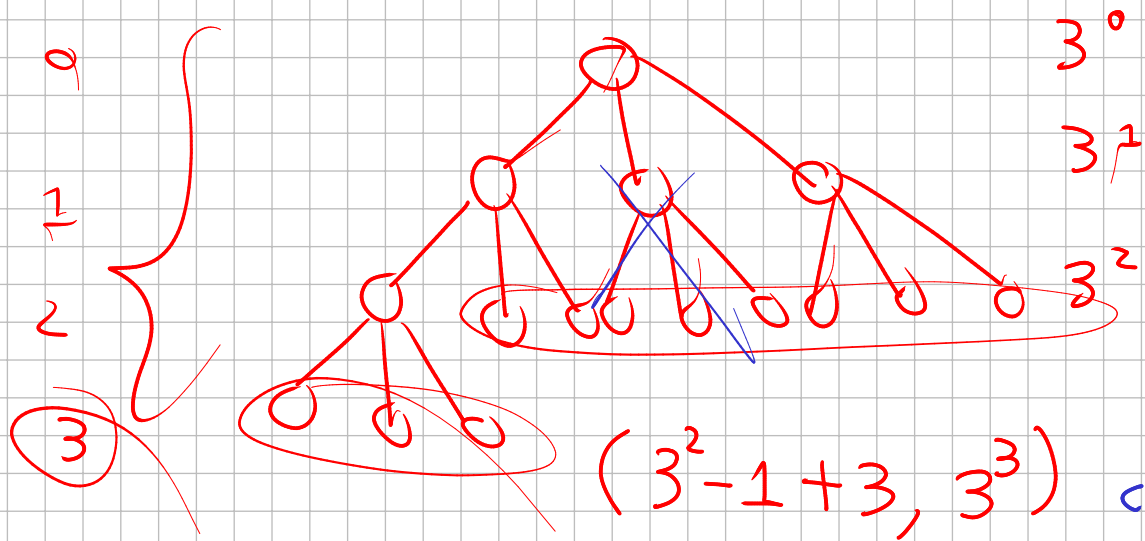
$$2^3 - 1 + 2$$

$$\boxed{2^3 + 1}$$

hojas $(m^{h-1} + m - 1, m^h)$

$(2^3 + 2 - 1, 2^4)$

$(2^3 + 1, 2^4)$



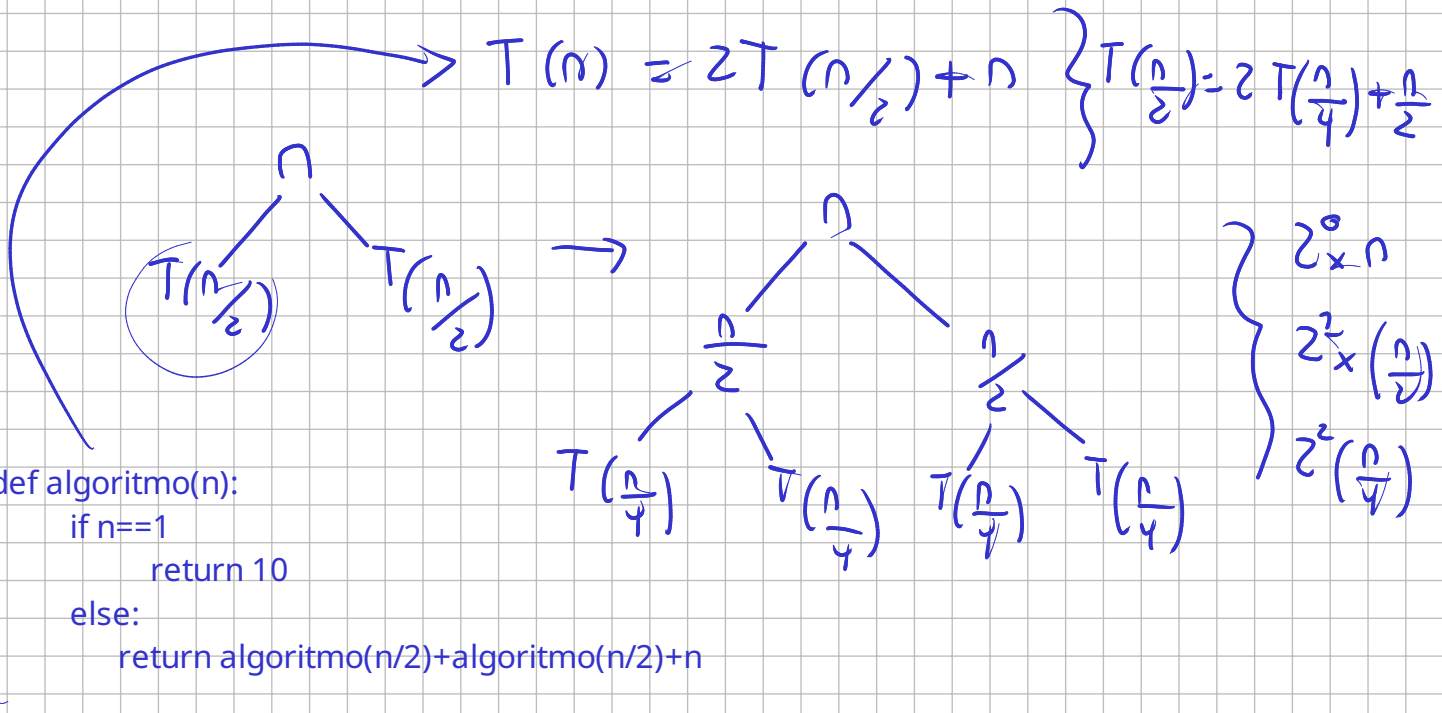
Completo y balanceado

$(3^2-1+3, 3^3)$

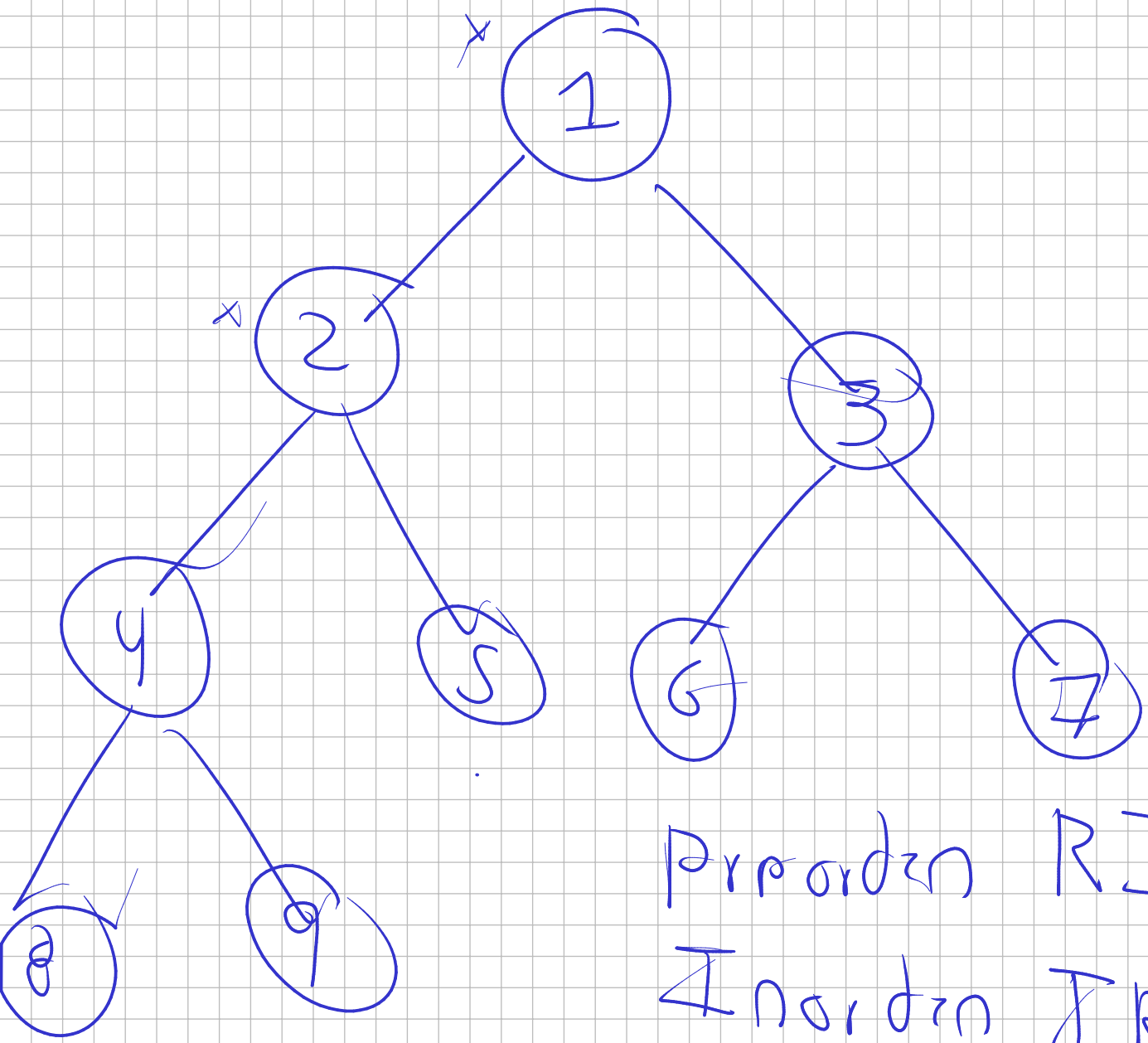
$(3^2+2, 3^3)$

completo nodo → n hijos

$(n^{h-1}-1+n, n^h)$



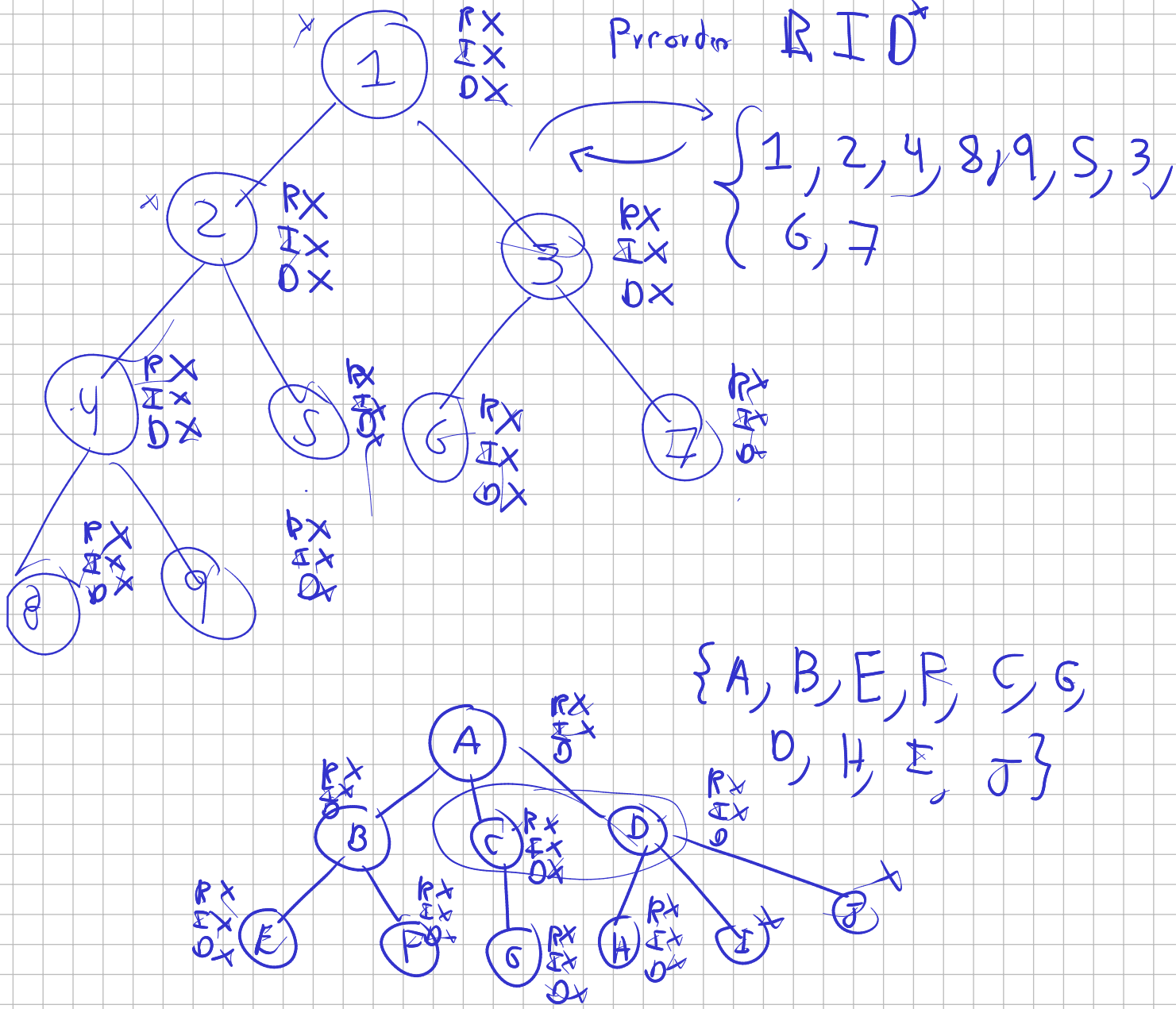

```
def algoritmo(n):
    if n==1
        return 10
    else:
        return algoritmo(n/2)+algoritmo(n/2)+n
```



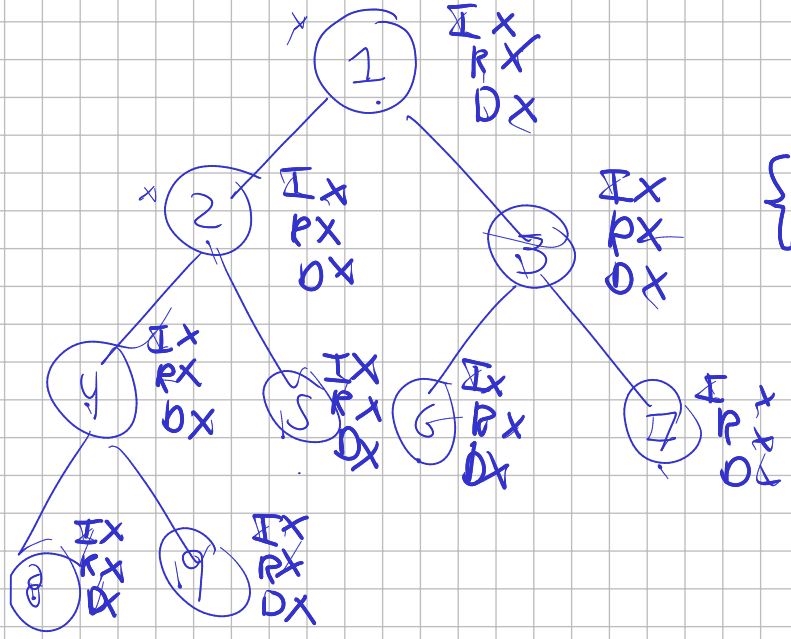
Preorden RID*

Inorden IRD*

Posorden IRD

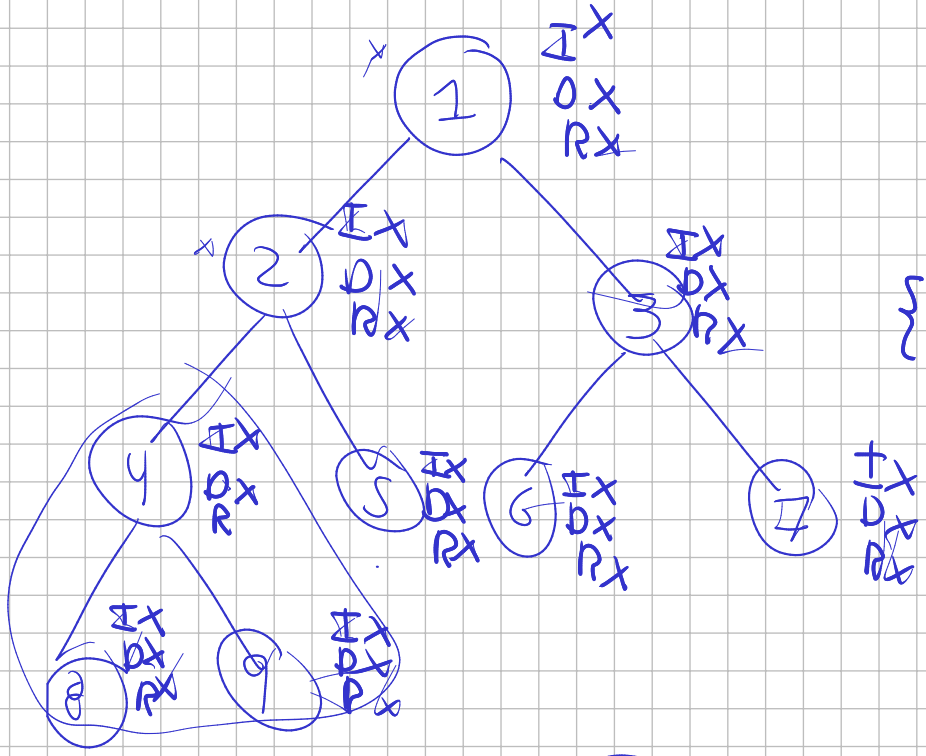

Preorden RID
{1, 2, 4, 8, 9, 5, 3, 6, 7}

{A, B, E, F, C, G, D, H, I, J}



Inorden I R D

{8, 4, 9, 2, 5, 1, 6, 3, 7}

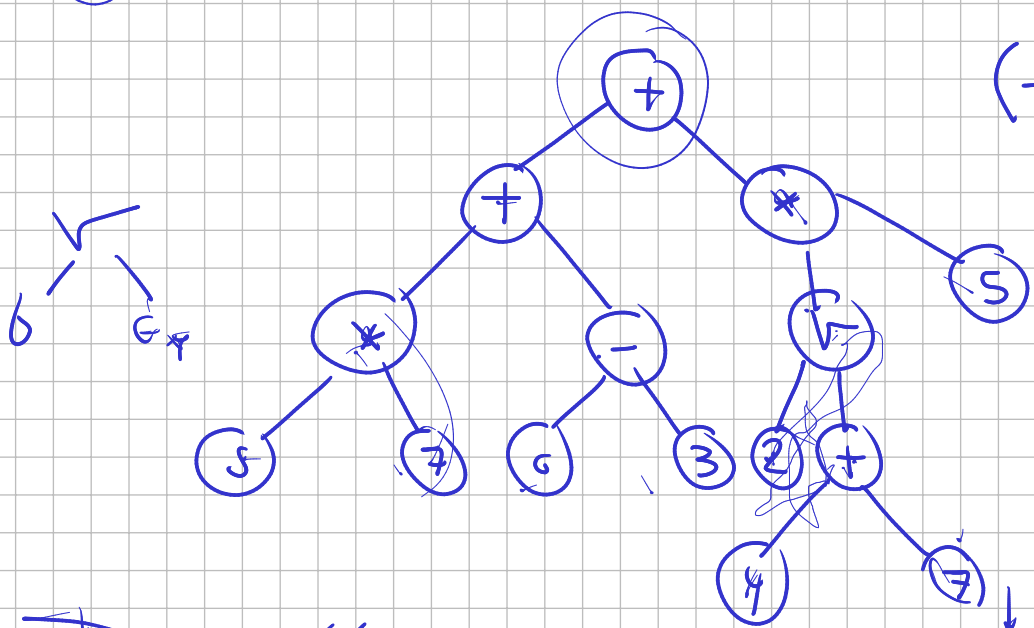


Posorden I, D, R

{8, 9, 4, 5, 2, 6, 7, 3, 1}

Preorden R I D

(+ (+ (* 5 7) (- 6 3)) (* (√ (+ 4 7)) 5))



Inorden ((5 * 7) + (6 - 3)) + $\sqrt[2]{(4+7)} * 5$

Postorden ((5 7 *) (6 3 -) +)

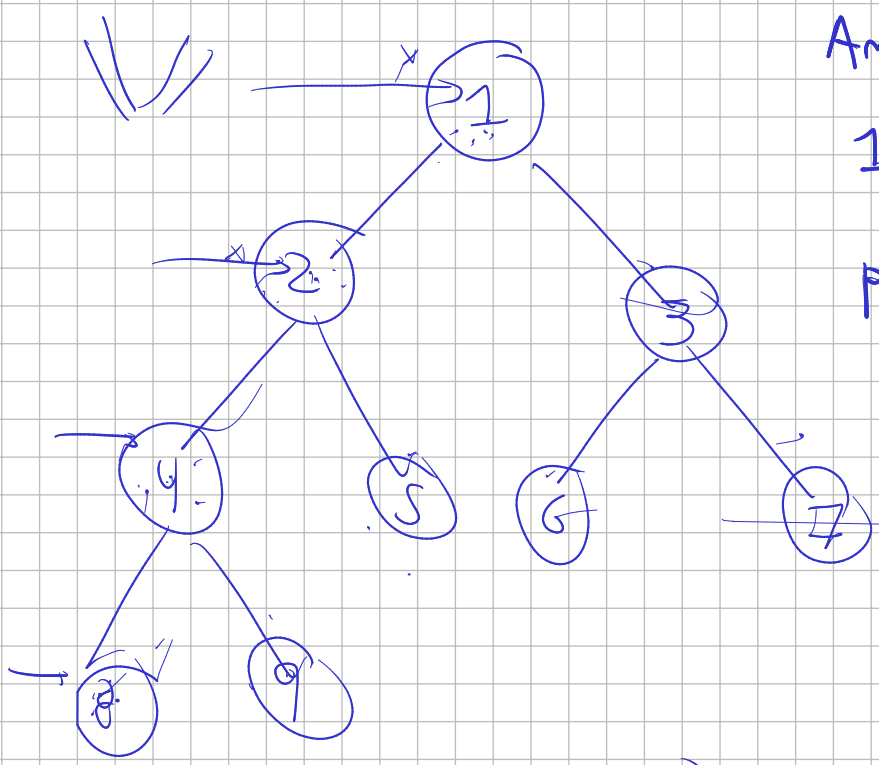
(((2 (4 7 +) /) 5 *) +)

(3 4 +)

LIFO

+ ←

3 4 4
3 3



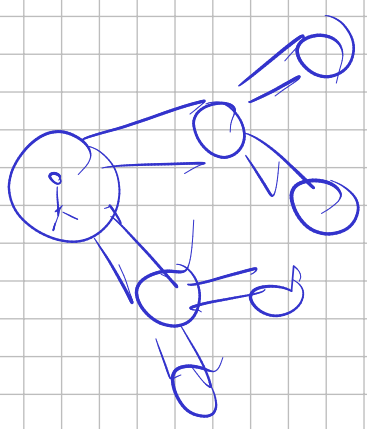
Amplitud

1, 2, 3, 4, 5, 6, 7, 8, 9

Profundidad I, D

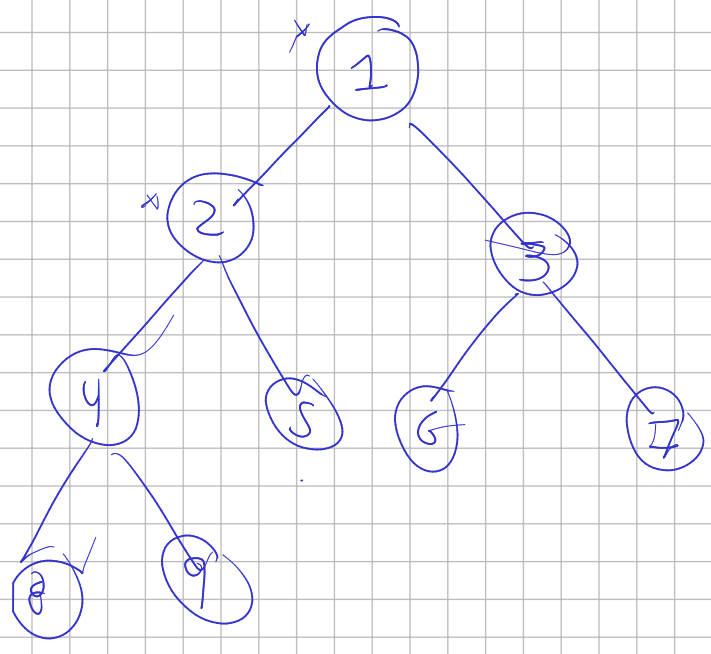
I 1, 2, 4, 8, 9, 5, 3, 6, 7

D 1, 3, 7, 6, 2, 5, 4, 9, 8



Amplitud cola

Profundidad pila



C = [1] S = {1}

C = [3] S = {1, 2}

C = [4 5] S = {1, 2 3}

C = [5 6 7] S = {1, 2, 3, 4}

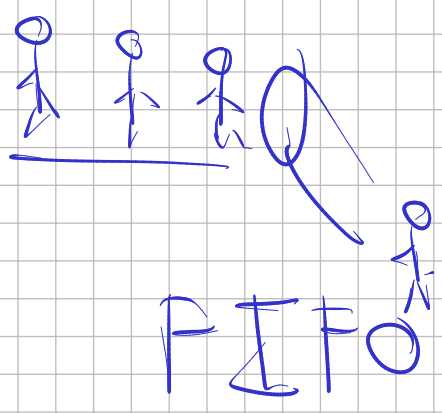
C = [6 7 8 9] S = {1, 2, 3, 4 5}

C = [7 8, 9] S = {1, 2, 3, 4, 5, 6}

C = [8, 9] S = {1, 2, 3, 4, 5 6, 7}

C = [9] S = {1, 2, 3, 4, 5 6, 7, 8}

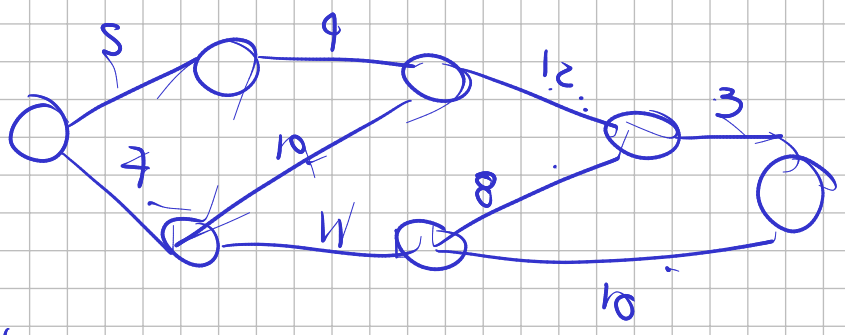
C = [ ] S = {1, 2, 3, 4 5, 6, 7, 8, 9}



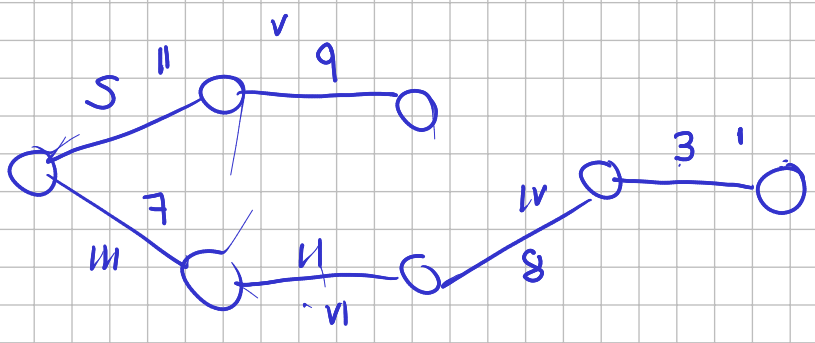
Arboles de expansión

Grafo -> Arbol: Eliminar los ciclos de un grafo para que sea un arbol

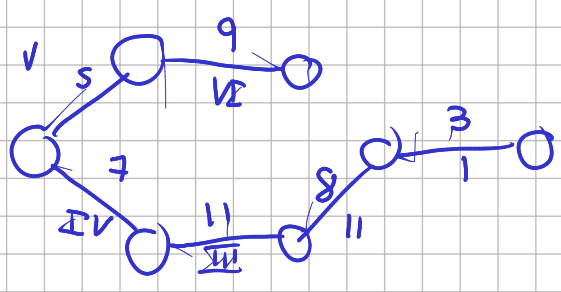
Arboles de expansión minimos: Cuya suma de los valores de las aristas sea mínimo

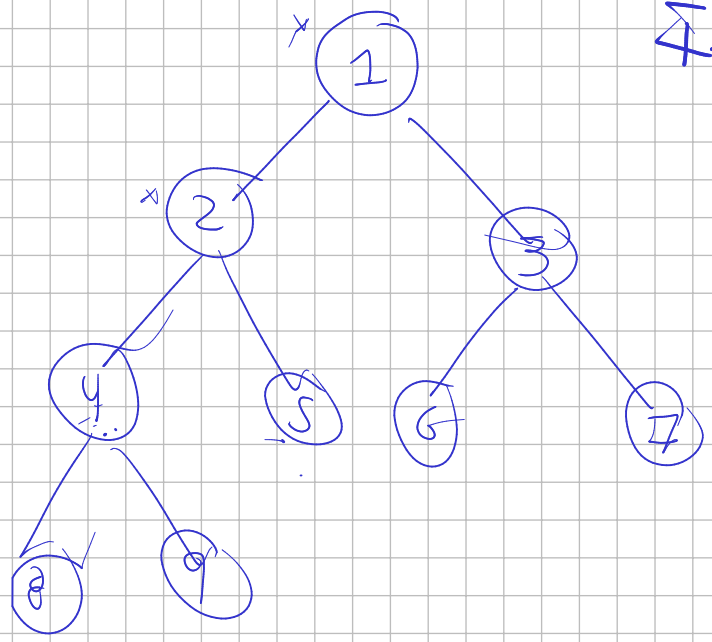


Kruskal



Prim





Prof

P = [1] S = { }

P = [2 3] S = {1}

P = [4 5 3] S = {1, 2}

P = [8 9 5 3] S = {1, 2, 4}

P = [9 5 3] S = {1, 2, 4, 8}

P = [5 3] S = {1, 2, 4, 8, 9}

P = [3] S = {1, 2, 4, 8, 9, 5}

P = [6 7] S = {1, 2, 4, 8, 9, 5, 3}

P = [7] S = {1, 2, 4, 8, 9, 5, 3, 6}

P = [ ] S = {1, 2, 4, 8, 9, 5, 3, 6, 7}