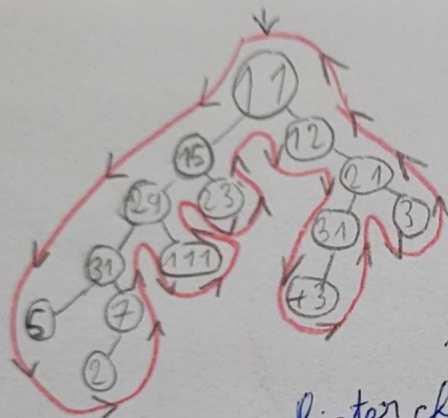


# *Parcourir un Arbre*

1)



Parcours préfixe:

11; 15; 29; 31; 5; 7; 2; 11; 23; 12; 21;

31; 43; 3

11 → 5 puis on remonte ~~avec~~ 7 → 2 11 23 12 → 43 et 3

lister chaque sommet première fois qu'on se rencontre

VARIABLE

T : arbre

x : noeud

DEBUT

PARCOURS-PREFIXE(T) :

si T ≠ NIL :

x ← T.racine

affiche x.clé

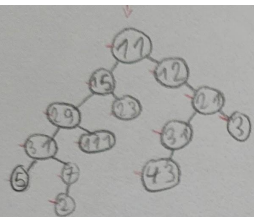
PARCOURS-PREFIXE(x.gauche)

PARCOURS-PREFIXE(x.droit)

fin si

FIN

2)



Parcours infixe

6; 31; 2; 7; 29; 111; 15; 23; 11; 43; 31; 21; 3; 12  
1 2 3 3 2 3 2 3 2 1 1 2 3 2

1 = infixe (fils gauche de l'arbre)

2 = racine de l'arbre

3 = infixe ( fils droit de l'arbre )

algorithme parcours infixe

## Variable

T: arbre

sc: noed

DEBUT

PARCOURS-INFIXE (T):

si  $T \neq \text{NIL}$ :

$\alpha \leftarrow T. \text{racine}$  1<sup>re</sup> gauche

PARCOURS-INFIXE (fils gauche de l'arbre)

afficher la racine de l'arbre

PARCOURS-INFIXE (fils droit de l'arbre)

fin si

FIN

algorithm parcour suffixe

VARIABLE

T: arbre

PL: no end

DERUT

PARCO

$\text{Ti} \neq \text{NIL}$

706-

DARCOUBS SUD

PARCOURS SUITE (plus grande version)

PARCOURS. SUITE IXE (film écrit de l'art de...)

Effacer la racine de l'arbre

fiava si

FIN

 Complétez l'algorithme du parcours infixe ci-dessous :

VARIABLE

T : \_\_\_\_\_

x : \_\_\_\_\_

DEBUT

PARCOURS-INFIXE(T) :

si \_\_\_\_\_ :

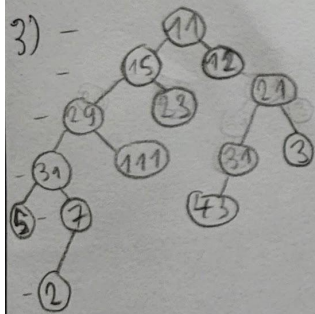
X ← .....

PARCOURS-INFIXE(\_\_\_\_\_)

PARCOURS-INFIXE( )

fin si

FIN



Parcours suffixe:

lister chaque sommet la dernière fois  
qu'on le rencontre

5; 2; 7; 31; 111; 29; 23; 15; 43; 31; 3; 21; 12; 11  
 1 2 2 3 2 3 2 3 1 1 2 3 3 3

1 = suffixe (fils gauche)

2 = suffixe (fils droite)

3 = racine de l'arbre