

Quelques exercices sur les arbres binaires

Exercice1 : reconstruction d'un arbre

Avec la déforestation massive, il est important de savoir reconstruire les arbres

Le but de cet exercice est de reconstruire un arbre connaissant son parcours infixe et son parcours suffixe.

a) Arbre 1

Le parcours infixe est [4, 8, 2, 5, 1, 6, 3, 7]

le parcours suffixe est [8, 4, 5, 2, 6, 7, 3, 1]

b) Arbre 2

Le parcours préfixe est E F T G S A V I J U

Le parcours infixe est G T F S E I V J A U

Dessiner ci-contre les 2 arbres correspondants :

a)

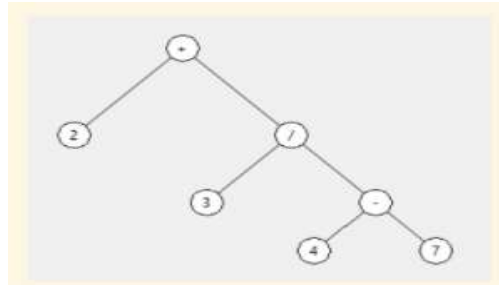
b)

Exercice2 : évaluer une expression

arithmétique :

Voici une expression arithmétique : $2 + \frac{3}{4-7}$

On peut la représenter par un arbre binaire



fonction `evaluer(arbre)`

Si

Sinon

Sinon

Sinon

Sinon

a) Ecrire ci-contre l'algorithme de la fonction `evaluer(arbre)` qui retourne la valeur de l'expression :

b) Implémenter -le dans le notebook avec la classe `Nœud`

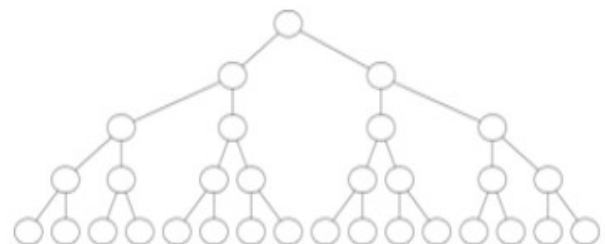
on définira l'arbre ci-dessus par

```
Arbre = Noeud('+', Noeud(2), Noeud('/', Noeud(3), Noeud('-', Noeud(4), Noeud(7))))
```

Exercice3 : Arbre binaire parfait

Un arbre binaire parfait est un arbre binaire dont chaque nœud (sauf les feuilles) possède exactement deux enfants. Toutes les feuilles sont à la même profondeur.

Voici ci-contre un arbre binaire parfait de hauteur 4 :



Ecrire la fonction `parfait(h)` qui prend en paramètre la hauteur de l'arbre souhaité et qui renvoie un arbre parfait de hauteur `h`.

Aide : la fonction est récursive.