NSI · TERMINALE / COURS

Les arbres binaires

Astrolab Cours · astrolabcours.fr

01 Le problème

Un tournoi à élimination directe, l'organisation des dossiers d'un ordinateur, une expression comme $(3 + 4) \times 2$: ces données ne sont pas rangées en ligne, elles se ramifient. Une liste ne suffit plus.

01 L'idée : l'arbre généalogique à l'envers

Pense à un arbre généalogique dessiné à l'envers : un ancêtre en haut, chacun a au plus deux enfants en dessous. Pour connaître toute la famille, il suffit de connaître l'ancêtre et les familles de ses deux enfants.

01 Le vocabulaire

Un **arbre binaire** est soit **vide**, soit un **nœud** (la **racine**) qui porte une valeur et deux sous-arbres binaires : le **sous-arbre gauche** et le **sous-arbre droit**.

- Une **feuille** est un nœud dont les deux sous-arbres sont vides.
- La **taille** est le nombre de nœuds.
- La **hauteur** est le nombre de nœuds sur le plus long chemin de la racine à une feuille (convention : arbre vide de hauteur 0, racine seule de hauteur 1).

En Python

```python
class Noeud:
    def __init__(self, valeur, gauche=None, droit=None):
        self.valeur = valeur
        self.gauche = gauche
        self.droit = droit

# l'expression (3 + 4) * 2
a = Noeud("*", Noeud("+", Noeud(3), Noeud(4)), Noeud(2))
```

L'arbre vide est représenté par `None`.

Taille et hauteur, en récursif

La définition est récursive, les fonctions aussi :

© Astrolab Cours · astrolabcours.fr/nsi/terminale/arbres-binaires/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

```python
def taille(a):
    if a is None:
        return 0
    return 1 + taille(a.gauche) + taille(a.droit)

def hauteur(a):
    if a is None:
        return 0
    return 1 + max(hauteur(a.gauche), hauteur(a.droit))
```

Les parcours

Parcours	Ordre	Sur l'arbre de $(3+4)\times 2$
préfixe	racine, gauche, droit	* + 3 4 2
infixe	gauche, racine, droit	3 + 4 * 2
suffixe	gauche, droit, racine	3 4 + 2 *
en largeur	niveau par niveau	* + 2 3 4

Les trois premiers sont récursifs ; le parcours en largeur utilise une file.

Exemple guidé

Avec l'arbre `a` de l'expression $(3+4)\times 2$, écrire une fonction qui calcule sa valeur.

```python
def evalue(a):
    if a.gauche is None and a.droit is None:
        return a.valeur          # une feuille : un nombre
    g = evalue(a.gauche)
    d = evalue(a.droit)
    if a.valeur == "+":
        return g + d
    return g * d

evalue(a)   # 14
```

1. **Cas de base** : une feuille contient un nombre, on le renvoie.
2. **Cas récursif** : on évalue les deux sous-arbres, puis on applique l'opérateur de la racine. C'est un parcours **postfixe** : les enfants d'abord, la racine ensuite.
3. Les trois parcours donnent : préfixe `* + 3 4 2`, infixe `3 + 4 * 2`, postfixe `3 4 + 2 *`. L'infixe perd les parenthèses, c'est pour ça qu'on les rajoute à l'affichage.

Retour au problème

L'arbre garde la structure que la liste écrasait : qui joue contre qui dans le tournoi, quel dossier contient quoi, quelle opération se fait en premier.

Les pièges

© Astrolab Cours · astrolabcours.fr/nsi/terminale/arbres-binaires/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

- La convention de hauteur varie selon les sujets : la lire dans l'énoncé.
- Gauche et droit ne sont pas interchangeables : deux arbres miroirs sont différents.
- Oublier le cas de l'arbre vide dans une fonction récursive.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/arbres-binaires/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits