

Algorithmique – Programmation Objet – Python

TD n° 11

Arbres

Licence Informatique 2ème année
Université de Nice-Sophia Antipolis

Un arbre T est une structure récursive composée par un nœud et ses sous-arbres.
On rappelle que:

- un nœud est la racine si ce nœud n'a pas de parents ;
- un nœud est une feuille si ce nœud n'a pas de fils ;
- un nœud v est un descendant d'un nœud u si u appartient au chemin de la racine vers v dans l'arbre ;
- un nœud u est un ancêtre d'un nœud v si u appartient au chemin de la racine vers v dans l'arbre.
- la profondeur d'un nœud u est la distance de la racine au nœud u .

1 Parcours

Dessinez 2 arbres binaires (chaque nœud interne possède exactement deux fils) ayant chacun 11 nœuds désignés par les lettres de a à k et dont la racine est a . Pour chacun de ces arbres, donnez l'ordre de parcours des nœuds suivant qu'il est préfixé, infixé ou postfixé.

2 Une classe ARBRE récursive

On supposera que l'on dispose d'une classe ARBRE dotée d'un constructeur ARBRE(ARBRE x), qui crée un nœud avec parent x , et les attributs suivants :

- FILS[] : un tableau contenant les références à des objets de classe ARBRE (les sous-arbres), vide si l'objet est une feuille. On supposera que la taille de ce tableau soit donnée par une méthode FILS.TAILLE() ;
- PARENT : référence au nœud parent, NIL si l'objet est la racine de l'arbre.
- CONTENU : référence à l'objet qui contient les données associées avec le nœud représenté par l'objet.

Écrivez les méthodes suivantes :

1. RACINE() : renvoie la racine de l'arbre dont l'objet fait partie.
2. NBDESCENDANTS() : renvoie le nombre de descendants d'un nœud.
3. NBFEUILLES() : renvoie le nombre de feuilles de l'arbre qui a le nœud comme racine.
4. PROFONDEUR() : renvoie la profondeur du nœud.
5. NBPETITFILS() : renvoie le nombre de petit-fils du nœud.
6. NBDESCENDANTS(k) : renvoie le nombre de descendants du nœud jusqu'à la k ème génération.
7. ESTDESCENDANT(x) : renvoie vrai si x est un descendant du nœud et faux autrement.

3 Ancêtre commun le plus proche

L'ancêtre commun le plus proche de deux nœuds u et v est le nœud w qui est un ancêtre de u et de v et qui est le plus profond.

1. Dessinez un arbre et recherchez l'ancêtre commun le plus proche de deux nœuds.
2. Écrivez un algorithme qui renvoie cet ancêtre commun le plus proche. Aide : on remonte d'un nœud vers la racine et on marque les ancêtres rencontrés. On fait de même pour le second nœud, on montre alors que le premier nœud marqué rencontré est le nœud recherché.

4 Arbres binaires

Dessinez un arbre non binaire, montrez qu'il peut être représenté sous la forme d'un arbre binaire.

5 Évaluation de formules propositionnelles

En utilisant la classe ARBRE développée ci-dessus, supposons maintenant

- que le CONTENU de chaque feuille soit une lettre (par exemple p, q, \dots) qu'on appellera *symbole propositionnel* ;
- que le CONTENU de chaque nœud interne soit un des opérateurs logiques \vee (disjonction), \wedge (conjonction) et \neg (négation) ;
- qu'on dispose d'un objet \mathcal{I} , dit *interprétation*, qui expose une méthode VALEUR(x) qui renvoie la valeur de vérité (V ou F) de chaque symbole propositionnel x .

Un tel arbre est une représentation d'une formule en logique propositionnelle.

Écrivez la méthode VALEUR(\mathcal{I}) de la classe ARBRE qui renvoie la valeur logique de la formule propositionnelle représentée par l'arbre dont l'objet est la racine selon l'interprétation \mathcal{I} .