DIU « Enseigner l'Informatique au Lycee » - UE 5

ARBRES

Classe Arbre (binaire de recherche)

A partir des informations du cours, construire et tester une classe `Arbre` implémentant les arbres binaires de recherche, incluant :

- un constructeur et trois attributs info, fg et fd
- les méthodes estVide, insererElement, afficherParcoursInfixe, afficherParcoursLargeur, et rechercherElement vues en cours

Hauteur d'un arbre

Ecrire la méthode de la classe `Arbre` qui détermine sa hauteur. Pour rappel, cette méthode retournera -1 si l'arbre est vide.

Parcours préfixe itératif d'un arbre binaire

Ecrire une méthode **itérative** (et non récursive) de la classe `Arbre` qui affiche le parcours préfixe.

Egalité de deux arbres binaires

Ecrire une méthode de la classe `Arbre` qui teste l'égalité avec un arbre passé en paramètre.

Recherche itérative dans un arbre binaire quelconque

Ecrire une fonction **itérative** de la classe `Arbre` qui cherche un élément dans un arbre binaire quelconque. La méthode retournera un tuple dont le premier élément est un booléen indiquant si l'élément est présent et le deuxième le noeud de l'arbre où il est présent (None si absent).

Tester si un arbre binaire est un arbre binaire de recherche

Ecrire une fonction de la classe `Arbre` qui renvoie vrai si l'arbre est un arbre binaire de recherche et faux sinon.

Valeur maximale d'une branche

Nous souhaitons écrire une fonction de la classe `Arbre` qui renvoie la somme maximale des valeurs des nœuds depuis la racine vers n'importe quelle feuille de l'arbre (en ne visitant qu'une seule fois chaque noeud).

Donner une version gloutonne de cette fonction, où on fera localement le meilleur choix possible. Cet algortihme donne-t-il toujours la valeur optimale ? Quelle est sa complexité ?

La programmation dynamique est une technique pour résoudre des problèmes algorithmiques qui consiste à les séparer en sous-problèmes qui suivent une sous-structure optimale. Elle peut s'en servir pour calculer efficament des quantités définies récursivement. Sur un arbre, l'idée est d'associer une valeur à chaque noeud qui combine les valeurs de ces fils.

Donner une version en programmation dynamique (donc optimale) de l'algorithme précédent. Quelle est sa complexité ?

Suppression dans un ABR

Ecrire la méthode de la classe Arbre qui supprime un élément passé en paramètre. Attention, après suppression l'arbre devra toujours être un ABR.