

## EXERCICE 2 (4 points)

Cet exercice porte sur les arbres binaires de recherche.

**Les questions sont indépendantes.**

Dans cet exercice, on s'intéresse aux arbres binaires de recherche dont les clés des nœuds sont des nombres entiers tous différents deux à deux.

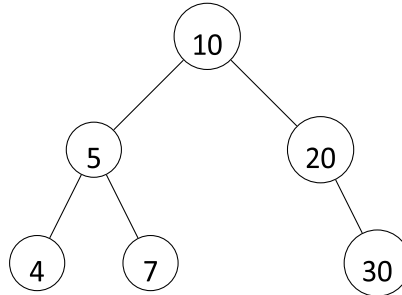


Figure 1 : un arbre binaire de recherche

**Rappel :** Un arbre binaire de recherche (ABR) est un arbre binaire étiqueté avec des clés tel que :

- Les clés du sous arbre gauche sont inférieures ou égales à celle de la racine ;
- Les clés du sous arbre droit sont strictement supérieures à celle de la racine ;
- Les deux sous arbres sont eux-mêmes des arbres binaires de recherche.

1. On rappelle que lors d'un parcours *en profondeur* d'un arbre binaire, on définit trois parcours des nœuds de l'arbre :
  - le parcours préfixe;
  - le parcours infixé;
  - le parcours suffixe (ou postfixé).

On parcourt l'arbre de la *figure 1 en profondeur*.

- (a) Quel est le parcours suffixe de cet arbre?
  - (b) Quel parcours permet d'avoir les clés des nœuds de l'arbre dans l'ordre croissant, c'est à-dire 4, 5, 7, 10, 20, 30 ?
2. On rappelle qu'un arbre ne comportant qu'un seul nœud a une hauteur égale à 1.
    - (a) Recopier l'arbre binaire de recherche de la *figure 1* après l'insertion de la clé 28.
    - (b) Quelle est la hauteur de l'arbre obtenu après cette insertion?

Dans la suite, on considère la structure de données abstraites ABR (Arbre Binaire de Recherche) que l'on munit des opérations suivantes :

Structure de données : ABR

Utilise : Booleen, Element

Opérations :

creer\_arbre :  $\emptyset \rightarrow$ ABR

creer\_arbre() renvoie un arbre vide

est\_vide : ABR $\rightarrow$ Booleen

est\_vide(a) renvoie True si l'arbre a est vide et False sinon

racine : ABR $\rightarrow$  Element

racine(a) renvoie la clé de la racine de l'arbre non vide a

sous\_arbre\_gauche : ABR $\rightarrow$ ABR

sous\_arbre\_gauche(a) renvoie le sous-arbre gauche de l'arbre non vide a

sous\_arbre\_droit : ABR $\rightarrow$ ABR

sous\_arbre\_droit(a) renvoie le sous-arbre droit de l'arbre non vide a

inserer : ABR, Element $\rightarrow$ Rien

inserer(a, e) insère la clé e dans l'arbre a.

3. On définit une fonction *mystere* qui a pour paramètre un ABR a.

```
1 def mystere(a) :
2     assert not est_vide(a), 'L'arbre est vide'
3     if est_vide(sous_arbre_droit(a)) :
4         return racine(a)
5     return mystere(sous_arbre_droit(a))
```

(a) Quelle valeur est renvoyée lors de l'appel de la fonction *mystere* avec l'arbre de la *figure 1* comme argument ?

(b) Que représente cette valeur ?

4. Recopier et compléter la fonction *rechercher* qui a pour paramètres un ABR a et une clé e, et qui renvoie True si la clé e est dans l'arbre et False sinon.

```
1 def rechercher(a, e) :
2     if est_vide(a) :
3         return # A compléter
4     elif # A compléter :
5         return True
6     elif e < racine(a) :
7         return # A compléter
8     else :
9         return # A compléter
```