LU2IN003
Arbres binaires (suite) - Exercices de base semaine 7

Exercice(s)

1 Arbres H-équilibrés

Exercice 1 – Arbres binaires et nombres de Fibonacci

On rappelle la définition des nombres de Fibonacci : $F_0 = 0$, $F_1 = 1$ et $F_k = F_{k-1} + F_{k-2}$ si $k \geq 2$.
On donne la définition inductive suivante des *arbres de Fibonacci* :
- $T_0 = \emptyset$, $T_1 = \bullet$
- $T_k = (\bullet, T_{k-1}, T_{k-2})$ si $k \geq 2$

Question 1
Dessiner T_2, T_3, T_4, T_5.

Question 2
Montrer que $n(T_k) = F_{k+2} - 1$, pour tout $k \in \mathbb{N}$ par induction structurelle.

Question 3
Montrez par induction structurelle que, pour tout $k \geq 0$, $h(T_k) = k$ et que T_k est H-équilibré.

2 Arbres parfaits et tas

Exercice 2 – Hauteur et taille d'un arbre parfait

Question 1
Montrer que la taille n d'un arbre parfait de hauteur h vérifie $2^{h-1} \leq n < 2^h$.

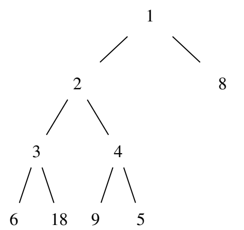
Question 2
En déduire la valeur de h en fonction de n. Quelle est la hauteur d'un arbre parfait de 10000 sommets ?

Exercice 3 – Opérations sur les tas

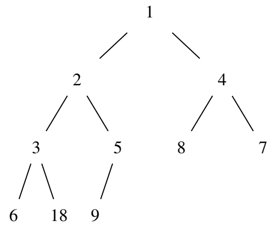
Dans cet exercice on considère des arbres binaires étiquetés par des nombres.
Un *tournoi* est un arbre binaire dont les étiquettes croissent de la racine vers les feuilles. Un *tas* est un tournoi parfait.
Le *parcours par niveau* d'un arbre parfait est la liste obtenue en parcourant les nœuds de l'arbre niveau par niveau et de gauche à droite.

Question 1
Pour chacun des trois arbres binaires T_1, T_2, T_3 suivants, dire s'il est parfait, s'il est un tas. Justifier les réponses.
On rappelle qu'un arbre parfait de taille n peut être représenté au moyen d'un tableau $A[0..N]$, avec $N \geq n$, tel que :

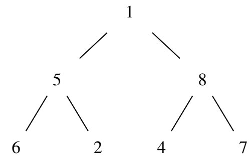
Arbre T_1 :



Arbre T_2 :



Arbre T_3 :

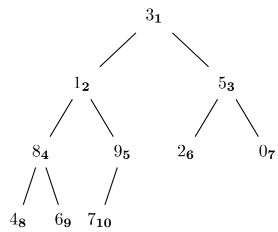


— $A[0]$ contient la taille n de T

— les cases $A[1..n]$ sont remplies en parcourant T de gauche à droite, niveau par niveau.

Ce tableau A est appelé *parcours par niveau* de l'arbre parfait T.

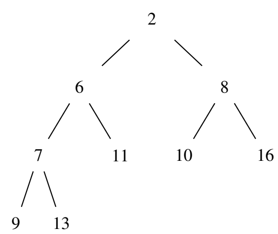
On peut numéroter les nœuds de l'arbre parfait T au moyen de leur position dans le tableau A comme illustré dans l'exemple suivant :



Le tableau associé à est $[10, 3, 1, 5, 8, 4, 9, 2, 0, 4, 6, 7]$

Question 2

On considère le tas ExT :



1. Donner le tableau ExA associé à ExT.
2. Réaliser l'insertion de la clé 5 dans le tas ExT, en maintenant la structure de tas. Décrire brièvement (deux phrases maximum) l'algorithme utilisé. Donner le tableau associé au résultat.
3. Réaliser la suppression de la clé minimale dans le tas ExT, en maintenant la structure de tas. Décrire brièvement (deux phrases maximum) l'algorithme utilisé. Donner le tableau associé au résultat.

Question 3

Soit L une liste de n entiers naturels.

1. Décrire brièvement (deux phrases maximum) l'algorithme permettant de trier la liste L en utilisant un tas. Quelle en est la complexité en nombre de comparaisons dans le pire cas ?
2. Soit $L_0 = (5, 2, 3, 1, 4)$.
   (a) Construire le tas T_0 obtenu en insérant successivement les éléments de L_0 et le tableau A_0 associé à ce tas.
   (b) Détruire le tas T_0 par suppressions successives du minimum. On dessinera l'arbre obtenu à chaque suppression et on donnera le tableau associé à chaque tas.

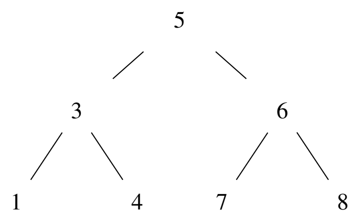
3 Arbres binaires de recherche (ABR)

Exercice 4 – Définition des ABR

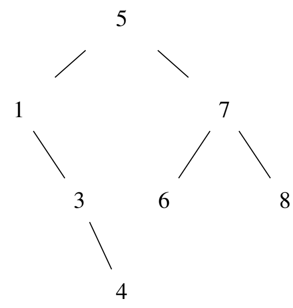
Question 1

Est-ce que les arbres B_1, B_2 et B_3 sont des ABR ? Justifier les réponses négatives.

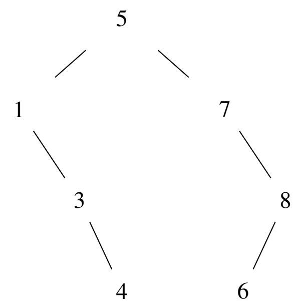
B_1



B_2



B_3



Exercice 5 – Caractérisation d'un ABR

On rappelle que, si T est un ABR, `ABinfixe(T)` est une liste rangée en ordre strictement croissant.

Question 1

1. Dessiner trois ABR différents dont les clefs sont $3, 1, 4, 8, 5, 2, 7, 6$. Donner le parcours infixe de chacun d'eux.
2. Pouvez-vous dessiner deux ABR différents ayant la même forme et les mêmes clefs (deux à deux distinctes) ?

Question 2

Montrer que, si le parcours infixe des clefs d'un arbre binaire T est rangé en ordre strictement croissant alors T est un ABR. Cette propriété peut être démontrée par récurrence sur la taille de l'arbre.

Question 3

1. Décrire en une phrase le principe d'un algorithme qui teste si un arbre binaire est un ABR.
2. En supposant que les listes sont représentées par des listes circulaires doublement chaînées, calculer la complexité de cet algorithme.

Question 4

Soit P une liste dont les éléments sont deux à deux distincts. Montrer que, s'il existe un ABR dont le parcours préfixe est P, alors cet arbre est unique.

Exercice 6 – Exemple de manipulation d'un ABR

Question 1

L'ajout de nouvelles clefs dans un ABR se fait par insertion aux feuilles. Construire l'ABR Tex obtenu par insertion successive des clefs 32, 7, 23,64, 18, 28, 41 et 58. Est-ce que cet ABR est unique ? Que se passe-t'il si l'on change l'ordre des clefs ?

Question 2

Donner l'ABR obtenu en supprimant le maximum de Tex.

Question 3

Donner l'ABR obtenu en supprimant la racine de Tex.

Question 4

Donner l'ABR obtenu en supprimant la clef 7 dans Tex.

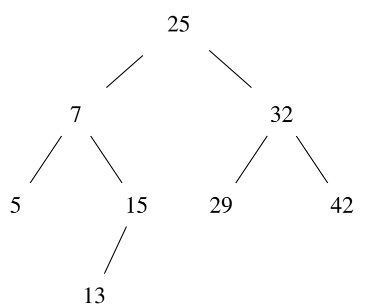
Exercice 7 – Insertion d'une clef dans un ABR

Dans cet exercice, on étudie l'algorithme d'insertion d'une clef dans un ABR rappelé à la suite :

```
def ABRinsertion(x,T):
  if estABvide(T):
    return ABfeuille(x)
  if x == T.clef:
    return T
  if x < T.clef:
    return AB(T.clef,ABRinsertion(x,T.gauche),T.droit)
  return AB(T.clef,T.gauche,ABRinsertion(x,T.droit))
```

Question 1

On considère dans cette question l'ABR suivant. Donner son parcours infixe. Conclusion ? Construire l'ABR obtenu après l'insertion de la clef 31. Donnez alors le parcours infixe de ce nouvel ABR.



Question 2

Montrer que `ABRinsertion(x,T)` se termine et retourne un ABR dont les clés sont x et les clés de T.