

Algorithmique

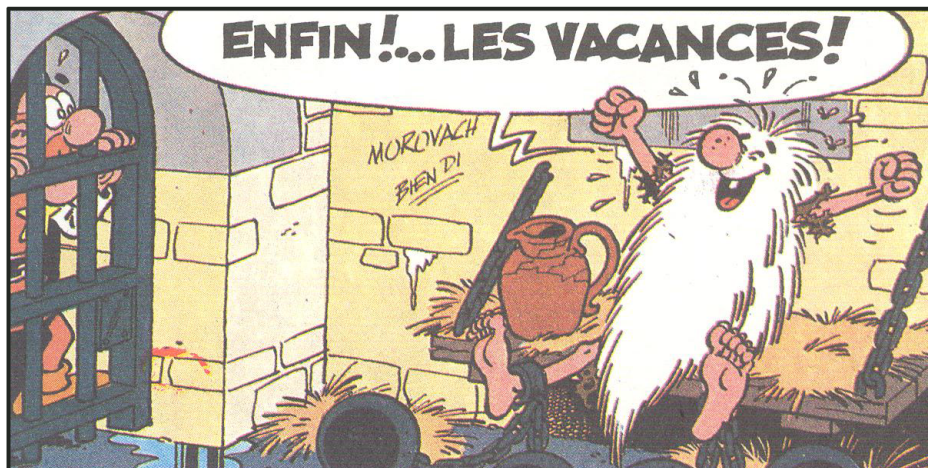
Partiel n° 2

INFO-SUP – EPITA

D.S. 311427.25 BW (5 juin 2012 - 09 :00)

Remarques (à lire !) :

- ☐ Vous devez répondre sur **les feuilles de réponses prévues à cet effet**.
 - Aucune autre feuille ne sera ramassée (gardez vos brouillons pour vous).
 - Répondez dans les espaces prévus, **les réponses en dehors ne seront pas corrigées** : utilisez des brouillons !
 - Ne séparez pas les feuilles à moins de pouvoir les ré-agrafer pour les rendre.
 - Aucune réponse au crayon de papier ne sera corrigée.
 - ☐ La présentation est notée en moins, c'est à dire que vous êtes noté sur 20 et que les points de présentation (2 au maximum) sont retirés de cette note.
 - ☐ **Les algorithmes :**
 - Tout algorithme doit être écrit dans le langage ALGO (pas de C#, CAML ou autre).
 - Tout code ALGO non indenté ne sera pas corrigé.
 - En dehors d'indication dans les énoncés, vous ne pouvez utiliser aucune routine (fonction ou procédure) supplémentaire.
 - Tout ce dont vous avez besoin (opérations de types abstraits, types) est donné en annexe.
 - **Rappel :** pour chaque algorithme, lorsque demandé sur les feuilles de réponses, vous devez donner :
 - **Les spécifications** c'est à dire ce qu'il fait, les paramètres (types et significations) et les éventuelles conditions d'utilisation.
 - **Le principe algorithmique** c'est à dire en clair la méthode retenue pour résoudre le problème. *Attention le principe et les spécifications sont notés, ne pas les envisager revient à sacrifier directement des points.*
 - ☐ Durée : 3h.
-



Exercice 1 (Arbres Bicolores et Arbres 2.3.4. – 6 points)

1. Dessiner l'arbre 2.3.4. représenté par l'arbre bicolore de la figure 1

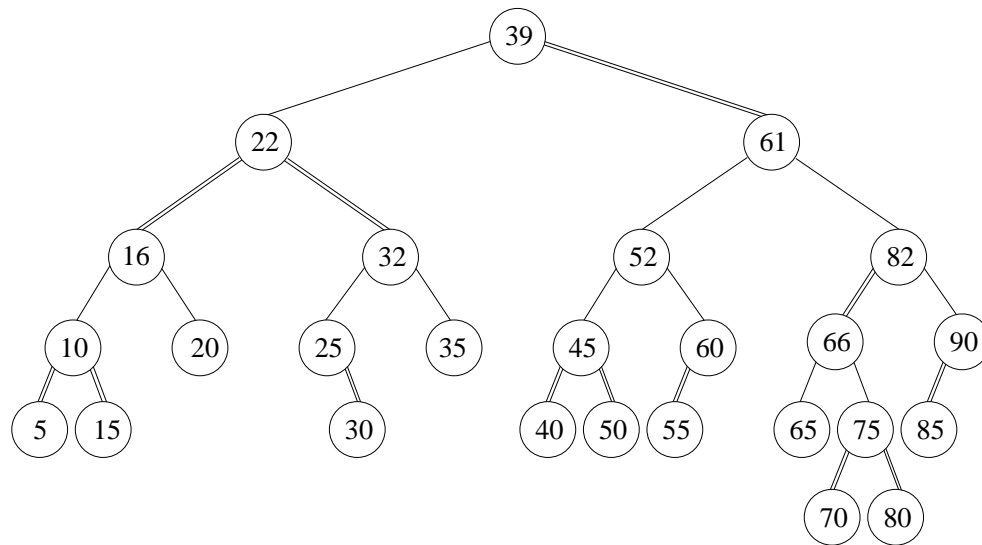


FIGURE 1 – Arbre Bicolore.

2. Quelle méthode **simple** permet de déterminer la hauteur d'un arbre 2.3.4. en utilisant l'arbre bicolore qui le représente? Justifier votre réponse.
3. Quelle méthode **simple** permet de déterminer la taille (le nombre de noeuds) d'un arbre 2.3.4. en utilisant l'arbre bicolore qui le représente? Justifier votre réponse.

Exercice 2 (AVL et mesures... - 9 points)

Effectuons un petit retour dans le temps pour parler des mesures sur les arbres et nous intéresser plus particulièrement à la hauteur d'un AVL.

1. Quel est le nombre maximum de noeuds d'un arbre AVL de hauteur h fixée? Justifiez votre réponse.
2. Soit T_h un AVL de hauteur h comportant le moins de noeuds possibles. Représenter graphiquement T_0 , T_1 , T_2 et T_3 .
3. On construit T_h (avec $h > 1$) en joignant un noeud x aux AVLs T_{h-1} et T_{h-2} . Démontrer que l'arbre obtenu est un AVL de hauteur h comportant le moins de noeuds possibles.
4. Notons $N(T_h)$ le nombre de noeuds de l'AVL T_h tel que défini à la question 3. En déduire une équation de récurrence linéaire F_h satisfaite par $F_h = 1 + N(T_h)$.

Remarques :

Les arbres utilisés dans les exercices suivants sont tous des arbres étiquetés (des entiers) de type `t_arbreBinaire` donné en annexe.

Exercice 3 (En long – 5 points)

On dispose d'un arbre binaire de recherche construit "à l'envers" : en chaque nœud de l'arbre, les valeurs contenues dans le fils droit sont inférieurs à la valeur de la racine, ceux du fils gauche lui sont supérieurs. On désire donc en faire une copie "à l'endroit", et détruire l'arbre initial.

Écrire l'algorithme correspondant.

Exercice 4 (En large – 5 points)

Compléter l'algorithme donné sur les feuilles de réponses pour qu'il calcule la hauteur et le nombre de feuilles d'un arbre binaire.

Exercice 5 (En travers... – 5 points)

On cherche ici à trouver le nombre de valeurs comprises entre deux bornes *inf* et *sup* dans un arbre binaire de recherche.

Écrire la fonction correspondante.

Annexes

Implémentation dynamique des arbres binaires

```
types
  t_arbreBinaire = ↑ t_noeudBinaire
  t_noeudBinaire = enregistrement
    entier      cle
    t_arbreBinaire fg, fd
fin enregistrement t_noeudBinaire
```

Routines autorisées

Nous supposons implémentées les opérations sur les files (type `t_file`) :

- `file_vide ()` : `t_file` : initialise la file
- `est_vide (t_file f)` : `booléen` : indique si *f* est vide
- `enfiler (t_elt_file e, t_file f)` : `t_file` : enfiler *e* dans *f*
- `defiler (t_file f)` : `t_elt_file` : défile et retourne le premier élément de *f*

Gestion de la mémoire

Il est bien entendu autorisé (souhaité...) d'utiliser les routines `allouer` et `liberer`.