Algorithmique 1

DS 3

Documents **non** autorisés
Remarque la notation dépendra de la lisibilité de la copie.

Exercice 3.1

Soit le tableau d'entiers `Tab={1, 12, 3, 5, 8 , 17, 9, 11, 4, 13}`
et la fonction `remplirTableauArbre` vue en cours (annexe B). Dessiner l'arbre binaire produit par l'appel à `remplirTableauArbre(Tab)`.

Exercice 3.2

```
fonction test1(val s:sommet): booleen;
debut
     si est_feuille(s) ou (filsGauche(s)!= nil et filsDroit(s)== nil) ou
         (filsGauche(s)== nil et filsDroit(s)!= nil) alors
         retourner faux;
    sinon retourner vrai;
    fsi
fin

fonction test2(val s:sommet): booleen;
debut
     si est_feuille(s) ou (filsGauche(s)!= nil et filsDroit(s)== nil)
         alors
               retourner vrai;
    sinon retourner faux;
    fsi
fin

fonction mystere(val A:arbreBinaire): booleen;
var F: Pile de sommet;
    s: sommet;
    b1,b2: booleen;
    c,h: entier;
debut
     creeFile(F);
     enfiler(F,A);
     c=1; b1=vrai;
     h=hauteur(A);
     Tant que (c <= 2**h -1  et b1==vrai et !fileVide(F)) faire
```

```
            s= valeur(F);
            defiler(F);
            si !test1(s) alors b=faux;
            sinon
                  enfiler(F,filsGauche(s));
                  enfiler(F,filsDroit(s)));
                  c=c + 2;
            fsi
      ftq
      si c < 2**h - 1  alors retourner faux;
      sinon
           b2=faux;
           Tant que (b1==vrai et !fileVide(F)) faire
                  s= valeur(F);
                  defiler(F);
                  si !b2 et test2(s) alors b2=vrai;
                  sinon si b2 et test2(s) alors b1=faux;
                  fsi
           ftq
       retourner b1;
fin
```

1. Faire la trace de l'exécution de `mystere` sur l'arbre construit dans l'exercice précédent, en donnant la valeur de chaque variable.
2. Que fait cette fonction en générale, justifier votre réponse.
3. Expliquer le rôle de la première boucle Tantque dans mystere.
4. Expliquer le rôle de la deuxième boucle Tantque dans mystere.
5. Quelle est la complexité de `mystere` ?

Exercice 3.3

1. Dessiner les trois premiers arbres binaire de recherche de racine 5 pour l'ensemble de clé $\{1, 3, 5, 7, 9, 11\}$.
2. Combien il y en a au total ? Justifier de manière succinte votre réponse.
3. Dans les mots infixes, caractériser le rang de la racine.
4. Ecrire une fonction qui renvoie le k^{ieme} élément d'un arbres binaire de recherche avec $k \leq n$ et $n =$ taille de l'arbre. Par exemple, la fonction appliquée à l'arbre dessiné ci-dessus, pour $k = 4$ elle renvoie 7.
5. Ecrire une fonction qui renvoie l'élément médian d'un arbres binaire de recherche. Par exemple, la fonction appliquée à l'arbre dessiné ci-dessus, elle renvoie 5.

Annexe A : Type abstrait arbreBinaire.

```
arbreBinaire= curseur;
sommet= curseur;
```

Primitives.

- Accès `fonction valeur(val S:sommet):objet;`
  /* vaut NIL si le sommet n'existe pas /*
  `fonction filsGauche(val S:sommet):sommet;`
  /* vaut NIL si S n'a pas de fils gauche */
  `fonction filsDroit(val S:sommet):sommet;`
  /* vaut NIL si S n'a pas de fils droit */
  `fonction pere(val S:sommet):sommet;` /* vaut NIL si S est la racine de l'arbre */
- Modification `fonction setValeur(ref S:sommet, val x:objet):vide;`
  /* affecte au sommet S la valeur x */
  `fonction ajouterFilsGauche(ref S:sommet, val x:objet):vide;`
  /* filsGauche(S)==NIL doit etre verifie */
  `fonction ajouterFilsDroit(ref S:sommet, x:objet):vide;`
  /* filsDroit(S)==NIL doit etre verifie */
  `fonction supprimerFilsGauche(ref S:sommet):vide;`
  /* filsGauche(S) est une feuille */
  `fonction supprimerFilsDroit(ref S:sommet):vide;`
  /* filsDroit(S) est une feuille */
  `fonction detruireSommet(ref S:sommet):vide;`
  /* S est une feuille */
- Création `fonction creerArbreBinaire(val Racine:objet):sommet;`

Annexe B : Construction d'un arbre binaire à partir d'un tableau d'entiers.

```
fonction remplirTableauArbre(ref T:tableau[1..N] d'entier):
                arbreBinaire d'entier;
  var A:arbreBinaire d'entier;
  var F: file de sommet;
  var s:sommet;
  var i:entier;
  début
    créerFile(F);
    A= creerArbreBinaire(T[1]);
    enfiler(F, A);
    tmp= 2;
    tantque 2*tmp-1 <= N faire
      pour i= tmp a 2*tmp-1 par pas de 2 faire
        s= getValeur(F);
        defiler(F);
        ajouterFilsGauche(s, T[i]);
        enfiler(F, filsGauche(s));
        ajouterFilsDroit(s, T[i+1]);
        enfiler(F, filsDroit(s));
      finpour;
      tmp= tmp*2;
    fintantque
    pour i=tmp à N-1 par pas de 2 faire
      s= getValeur(F);
      defiler(F);
      ajouterFilsGauche(s, T[i]);
      ajouterFilsDroit(s, T[i+1]);
    finpour;
    si N mod 2 == 0 alors
      ajouterFilsGauche(getValeur(F), T[N])
    finsi
    detruireFile(F);
    retourner(A);
  fin
```

Annexe C : Implémentation du type abstrait arbreBinaire.

```
cellule=structure
         info:objet;
         gauche: sommet;
         droit: sommet;
         pere: sommet;
       finstructure
sommet=^cellule;
arbreBinaire= sommet;

fonction creerArbreBinaire(val Racine:objet):sommet;
fonction valeur(val S:sommet):objet;
fonction filsGauche(val S:sommet):sommet;
fonction filsDroit(val S:sommet):sommet;
fonction pere(val S:sommet):sommet;

fonction setValeur(ref S:sommet, val x:objet):vide;

fonction ajouterFilsGauche(ref S:sommet, val x:objet):vide;
fonction ajouterFilsDroit(ref S:sommet, x:objet):vide;
fonction supprimerFilsGauche(ref S:sommet):vide;
fonction supprimerFilsDroit(ref S:sommet):vide;
fonction detruireSommet(ref S:sommet):vide;

fonction creerArbreBinaire(val Racine:objet):sommet;
```