**Algorithmique &Structures de données**

**Examen 1  // 2012 2013**

Corrigé

**{Algorithme de test**

 **Equilibrage d’un arbre de recherche binaire**

**}**

 **SOIT**

 **A UN ARB ;**

 **V UN VECTEUR ( 10 ) ;**

 **Transf1 UNE ACTION ;**

 **X UN ARB ;**

 **Parcours UNE ACTION ;**

 **Inserer UNE ACTION ;**

 **Preordre UNE ACTION ;**

 **Ind\_v UN ENTIER ;**

 **DEBUT**

 **CREER\_ARB ( A , [ 55 , 23 , 13 , 65 , 77 , 12 , 78 , 887 , 34 , 5 ] ) ;**

 **Ind\_v := 0 ;**

 **APPEL Transf1 ( A , V ) ;**

 **ECRIRE ( V ) ;**

 **X := NIL ;**

 **APPEL Parcours ( 1 , 10 ) ;**

 **APPEL Preordre ( X )**

 **FIN**

**{Insère la valeur Val dans l’arbre X}**

 **ACTION Inserer ( X , Val )**

 **SOIT**

 **X UN ARB ;**

 **Val UN ENTIER ;**

 **Trouv UN BOOLEEN ;**

 **P , Q , R DES POINTEURS VERS UN ARB ;**

 **DEBUT**

 **SI X = NIL**

 **CREERNOEUD ( X ) ;**

 **AFF\_INFO ( X , Val )**

 **SINON**

 **P := X ;**

 **Trouv := FAUX ;**

 **TQ ( P <> NIL ) ET NON Trouv :**

 **SI INFO ( P ) = Val**

 **Trouv := VRAI**

 **SINON**

 **Q := P ;**

 **SI Val < INFO ( P )**

 **P := FG ( P )**

 **SINON**

 **P := FD ( P )**

 **FSI**

 **FSI**

 **FTQ ;**

 **SI NON Trouv**

 **CREERNOEUD ( R ) ;**

 **AFF\_INFO ( R , Val ) ;**

 **SI Val < INFO ( Q )**

 **AFF\_FG ( Q , R )**

 **SINON**

 **AFF\_FD ( Q , R )**

 **FSI**

 **FSI**

 **FSI**

 **FIN**

**{Parcours Préordre de l’arbre A}**

 **ACTION Preordre ( A )**

 **SOIT**

 **A UN ARB ;**

 **DEBUT**

 **SI A <> NIL**

 **ECRIRE ( INFO ( A ) ) ;**

 **APPEL Preordre ( FG ( A ) ) ;**

 **APPEL Preordre ( FD ( A ) )**

 **FSI**

 **FIN**

**{Parcours l’arbre A en Inordre (T1 n T2) avec une pile pour construire un vecteur V}**

 **ACTION Transf1 ( A , V )**

 **SOIT**

 **A UN ARB ;**

 **V UN VECTEUR ( 10 ) ;**

 **P UN POINTEUR VERS UN ARB ;**

 **Une\_pile UNE PILE DE ARB ;**

 **Possible UN BOOLEEN ;**

 **DEBUT**

 **CREERPILE ( Une\_pile ) ;**

 **P := A ;**

 **Possible := VRAI ;**

 **TQ Possible**

 **TQ P <> NIL**

 **EMPILER ( Une\_pile , P ) ;**

 **P := FG ( P )**

 **FTQ ;**

 **SI NON PILEVIDE ( Une\_pile )**

 **DEPILER ( Une\_pile , P ) ;**

 **Ind\_v := Ind\_v + 1 ;**

 **AFF\_ELEMENT ( V [ Ind\_v ] , INFO ( P ) ) ; // ou bien V [ Ind\_v ] := INFO(P)**

 **P := FD ( P )**

 **SINON**

 **Possible := FAUX**

 **FSI**

 **FTQ**

 **FIN**

**{Parcours dichotomique du vecteur V pour construire l’arbre de racine X}**

 **ACTION Parcours ( Bi , Bs ) ;**

 **SOIT**

 **Bi , Bs , Mil DES ENTIERS ;**

 **DEBUT**

 **SI Bi <= Bs**

 **Mil := ( Bi + Bs ) / 2 ;**

 **APPEL Inserer ( X , ELEMENT ( V [ Mil ] ) ) ; // X et V globaux**

 **APPEL Parcours ( Bi , Mil - 1 ) ;**

 **APPEL Parcours ( Mil + 1 , Bs ) ;**

 **FSI**

 **FIN**

**5. Montrer que le nouvel arbre de recherche binaire ainsi construit est équilibré.**

**L’équilibrage est effectué en 2 étapes :**

**- On construit d’abord un tableau ordonné à partir des éléments de l’arbre car on a utilisé un parcours *inordre* de l’arbre.**

**- Le parcours dichotomique du tableau permet de récupérer les éléments dans l’ordre suivant pour N=10 : 5, 2, 1, 3, 4, 8, 6, 7, 9, 10**

**L’insertion des éléments du tableau dans un nouvel arbre selon l’ordre indiqué donne forcément un arbre équilibré.**

**Le 5ième élément est donc la racine de l’arbre. On insère ensuite le 2ième à gauche de la racine et le 1ier élément à gauche du 2ième. Le 3ième élément est inséré à droite du 2ième et le 4ième à droite du 3ième. Et ainsi de suite.**

**6. Imaginer un algorithme principal qui permet de tester la méthode d’équilibrage décrite.**

1. **Créer un arbre de recherche binaire en utilisant le module Insérer(A, v)**
2. **Faire un parcours *inordre*. Si tous les éléments sont affichés dans l’ordre, cela prouve que l’arbre est bien construit.**
3. **Appliquer la transformation Arbre 🡪 Tableau. Si le tableau contient tous les éléments de l’arbre et est trié, ceci est correct.**
4. **Appliquer la transformation Tableau 🡪 Arbre. L’insertion des éléments selon le parcours dichotomique du tableau devrait donner un arbre équilibré.**
5. **Une manière de vérifier que l’arbre est équilibré est d’afficher les éléments de l’arbre en préordre et de les ré insérer manuellement.**