Contrôle continu 3
Pointeurs - Récursivité – Listes – Piles – Files– Arbres
(3 pages)

Questions de cours (5 points)

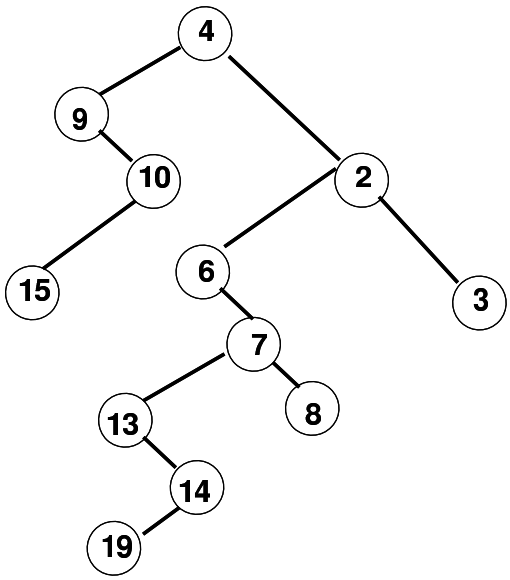
- Peut-on créer un arbre binaire dont la valeur du sommet est une liste?
- Utilise-t-on une file pour parcourir un arbre binaire en ordre infixe ? Si oui, pourquoi ? Si non, quelle structure de données est utilisée ?
- Quel sont les parcours qui existent pour les arbres binaires et pour les arbres planaires ?
- On considère la suite la suite d'opérations suivante :

```
var A : arbreBinaire de pile d'entier;
var s :pile d'entier;
creerPile(s) ;
empiler(s,0);
empiler (s,1);
A=creerArbreBinaire(s) ;
detruirePile(s))
 creerPile(s) ;
empiler(s,2) ;
ajouterFilsGauche(A,s)
empiler(s,3) ;
ajouterFilsDroit(A,s)
```

Dessinez la structure de donnée ainsi obtenue. Y-a-t-il des problèmes liés à l'implémentation de la pile ? en implémentation statique ? en implémentation dynamique ? On justifiera les réponses.

Exercice 2 (5 points)

Soit l'arbre binaire suivant



1. Donnez sa hauteur
2. Donnez la liste des sommets dans l'ordre postfixe (ou suffixe)
3. Donnez la liste des sommets dans l'ordre hiérarchique
4. Donnez l'arbre planaire correspondant dans la bijection de Knuth
5. Pour l'arbre planaire ainsi obtenu, donner la liste des entiers en ordre préfixe.
6.

Exercice 2 (10 points)

Ecrire une fonction qui renvoie la liste de tous les sommets internes d'un arbre binaire en ordre infixe :

1 – En utilisant les primitives du type arbreBinaire et listeSC

2 – En implémentant directement en allocation dynamique pour le type Arbre Binaire et le type listeSC.

Liste Simplement Chainées

fonction valeur(val L:liste d'objet):objet;
fonction debutListe(ref L:liste d'objet):vide;
fonction suivant(ref L:liste d'objet):vide;
fonction listeVide(val L:liste d'objet): booleen;
fonction créerListe(ref L:liste d'objet):vide;
fonction insérerAprès(ref L:liste d'objet; val x:objet;):vide;
fonction insérerEnTete(ref L:liste d'objet ;val x:objet):vide;
fonction supprimerAprès(ref L:liste d'objet):vide;
fonction supprimerEnTete(ref L:liste d'objet):vide;
fonction detruireListe(ref L:lisetd'objet):vide;</pre>

Liste Doublement Chainées (ajout de)

fonction finListe(ref L:liste d'objet):vide;
fonction précédent(ref L::liste d'objet): vide;

Fonctions sur les listes

fonction estFinListe(val L:liste d'objet):booléen;
fonction appartient(ref L:liste d'objet; ref x:objet): booléen ;

Piles

fonction valeur(ref P:pile de objet):objet;
fonction fileVide(ref P:pile de objet):booléen;
fonction créerPile(P:pile de objet);
fonction empiler(ref P:pile de objet;val x:objet):vide;
fonction dépiler(ref P:pile de objet):vide;
fonction detruirePile(ref P:pile de objet):vide;

Files

fonction valeur(ref F:file de objet):objet;
fonction fileVide(ref F:file de objet):booléen;
fonction créerFile(F:file de objet);vide;
fonction enfiler(ref F:file de objet;val x:objet):vide;
fonction défiler (ref F:file de objet):vide;
fonction detruireFile(ref F:file de objet):vide;

Arbres binaires

fonction getValeur(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 créerArbreBinaire(val Racine:objet):sommet;

Arbres planaires

fonction valeur(val S:sommetArbrePlanaire):objet;
fonction premierFils(val S:sommetArbrePlanaire):sommetArbrePlanaire;
fonction frere(val S:sommetArbrePlanaire):sommetArbrePlanaire;
fonction pere(val S:sommetArbrePlanaire):sommetArbrePlanaire;
fonction créerArborescence(val racine:objet):sommetArbrePlanaire;
fonction ajouterFils(ref S:sommetArbrePlanaire,val x:objet):vide;
fonction supprimerSommet(ref S:sommetArbrePlanaire):vide;
fonction créerArbreBPlaniare(val Racine:objet):sommet;