INF251 Algo 1
Devoir surveillé n° 3
45mn
Aucun document autorisé
30/11/2008

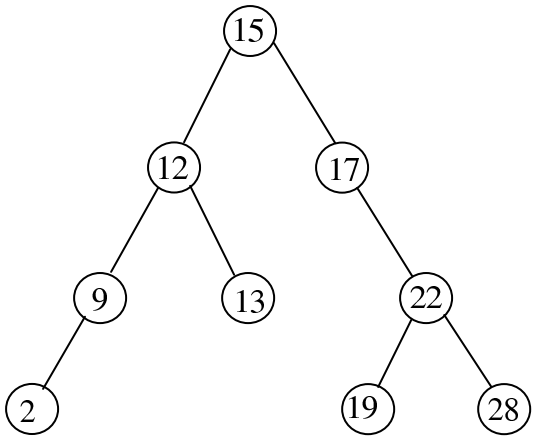
Exercice 1 :

1. Dessinez des arbres binaires de recherche de hauteur 2, 3, 4, 5 et 6 pour le même ensemble de valeurs {1, 4, 5, 10, 16, 17, 21}

2. Quelle est la complexité au pire de la fonction de recherche d'un élément dans un arbre binaire de recherche ? Justifiez

Exercice 2 :

Sans utiliser de structures de données supplémentaires écrire une fonction qui détermine le `k_ième_element` dans l'ordre croissant d'un arbre binaire de recherche



Sur l'arbre binaire de recherche `A` ci-dessus `k_ième_element(A,6)` renvoie `17` `k_ième_element(A,10)` renvoie `0`.

Exercice 3 :

On considère l'algorithme itératif de parcours en profondeur pour les arbres planaires ci-dessous :

```
Fonction ParcoursEnProfondeur (ref A: Arbre) :vide;
var x: Sommet;
P: PileDeSommet;
debut
      ViderPile(P);
      x:= Racine (A);
      <<1>>
      repeter
            Si ExisteFils(x)alors
                  <<2>>
                  Empiler(P,x);
                  x:= PremierFils(x)
                  <<3>>
            Sinon
                  <<4>>>
                  tant que ( x != Racine(A) et ! ExisteFrere(x)) faire
                        <<5>>
                        x:=ValeurPile(P); Depiler(P);
                        <<6>>
                  finTantque;
                  Si x <> Racine (A) alors
                        <<7>>
                        x:=Frere(x)
                        <<8>>
                  finSi
            finSi
      jusqu'a x = Racine(A)
      finRepeter
      <<9>>
fin;
```

Remarques : Les positions numériques sont les divers emplacements d'action à effectuer pendant le parcours.

1. Modifiez l'algorithme afin de compter le nombre de feuilles d'un arbre planaire

2. Soit MYST un tableau de `N` entiers. En considérant que l'arbre parcouru a au plus `N` sommets dont les valeurs sont des entiers $\in$`[1, N]` on modifie l'algorithme en ajoutant les instructions suivantes :

```
en <<1>> : pour tout i de 1 a N faire MYST[i]:= 0;
en <<2>>: MYST[getvaleur(x)] := MYST[getvaleur(x)] +1 ;
en <<7>> ou en <<8>> : MYST[getvaleur(PERE[x])] := MYST[getvaleur(PERE[x])] +1 ;
```

Pour un sommet x de valeur i que représentera la valeur MYST[i] à la fin du parcours ?

accès

```
fonction getValeur(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 être vérifié */
fonction ajouterFilsDroit(ref S:sommet,x:objet):vide;
/* filsDroit(S)==NIL doit être vérifié */
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 */
```