

IN325T2, Algorithmes et structures de données
Licence d'informatique
durée 1h30

Les notes de cours et TD sont autorisées.

Les deux parties A et B sont à rendre sur des copies séparées.

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

Partie A

Exercice 1. Tas

Question 1. *Rappeler la définition d'un tas (encore appelé file de priorité).*

Question 2. *A partir du tas vide, on insère successivement les éléments 7, 2, 6, 9, 8. Préciser, pour chaque insertion, le tas sous ses deux formes : arbre binaire et tableau.*

Exercice 2. Recherche du k -ème élément dans un arbre binaire de recherche (ABR)

On suppose que, pour chaque ABR considéré, toutes les valeurs des noeuds sont distinctes. Pour un entier k inférieur ou égal à la taille d'un ABR A , le k -ème élément de A est l'élément tel qu'il existe exactement $k-1$ éléments plus petits que lui dans A .

Dans cet exercice, on considère que tout ABR est représenté par la structure chaînée habituelle avec de plus le nombre de noeuds de son sous arbre gauche. Chaque noeud est ainsi composé de quatre champs :

- *val* la valeur du noeud;
- *fg* et *fd* les deux pointeurs sur les sous arbres gauche et droit;
- *tailleSAG* la taille du sous arbre gauche.

Question 3. *Ecrire en pseudo-code une fonction qui calcule le champ tailleSAG de chaque noeud de l'ABR A.*

Question 4. *Sans utiliser la fonction précédente, écrire une fonction qui insère un élément x dans un ABR A et qui met à jour les champs tailleSAG.*

Question 5. *Quelle est la complexité de cette fonction ? Justifier.*

Question 6. *Donner l'algorithme qui retourne le k -ème élément d'un ABR A, s'il existe.*

Partie B (A rendre sur une copie séparée)

Structure d'anneau

Certains algorithmes répartis utilisent un réseau structuré en anneau : chaque site connaît son successeur dans l'anneau, et chaque site du réseau fait partie de l'anneau. Ceci permet, en un tour de l'anneau, de visiter tous les sites du réseau une et une seule fois.

Question 7. *Citer un exemple d'algorithme réparti où le fait d'avoir une telle structure est très utile (justifier votre exemple).*

N.B. : lire la totalité du sujet peut être utile !

On considère l'algorithme suivant, basé sur un parcours séquentiel du réseau, et dont le but est de construire un anneau virtuel à partir d'un réseau quelconque. Un des sites, l'initiateur, joue un rôle particulier : il lance l'algorithme et, lorsque ce dernier sera terminé, pourra lancer un parcours dans l'anneau ainsi créé.

Les sites ont des identités deux à deux distinctes. Chaque site connaît ses voisins immédiats.

Variables :

Chaque site possède une variable successeur, une variable Pere, de type identité de site, et un tableau R d'identités de sites, indexé par les identités des sites du réseau. L'initiateur possède en outre une variable auxiliaire premier. Note : l'algorithme étant basé sur un parcours séquentiel, il n'y a qu'un message à la fois (la réception d'un message génère exactement un message, jusqu'à terminaison de l'algorithme).

```

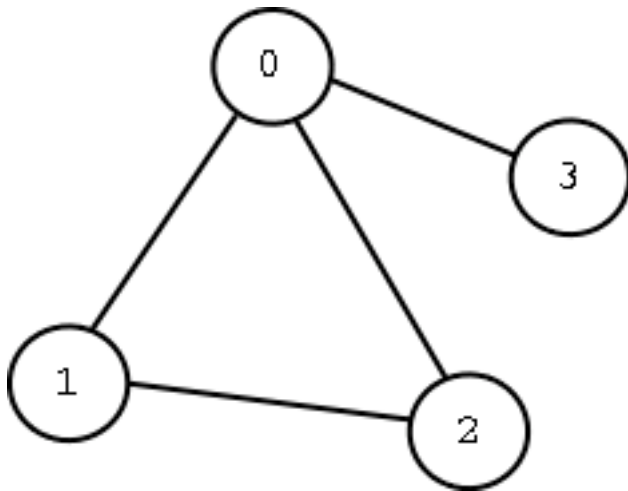
Site initiateur  $i_0$  :
    Choisit un voisin  $j$ 
    premier =  $j$ 
    Envoie explorer ( $\{i_0\}$ ,  $i_0$ ) à  $j$ 

Site quelconque  $i$  :
sur réception de explorer( $z$ , prec) venant de  $j$ 
     $z = z \cup \{i\}$ 
    Pere =  $j$ 
    successeur = prec
    si Voisins -  $z$  vide
        alors  $R[j] = j$ , et envoie rebrousser( $z$ ,  $i$ ) à  $j$ 
        sinon
            choisit  $k$  dans cet ensemble,
             $R[k] = j$ 
            envoie explorer( $z$ ,  $i$ ) à  $k$ 

sur réception de rebrousser( $z$ , prec) venant de  $j$ 
    si Voisins -  $z$  vide
        alors
            si Pere existe
                 $R[\text{Pere}] = j$ 
                envoie rebrousser( $z$ , prec) à Pere
            sinon
                successeur = prec
                 $R[\text{premier}] = j$ 
                fin de l'algorithme
                ( on est l'initiateur et le réseau est parcouru)
        sinon
            choisit  $k$  dans cet ensemble,
             $R[k] = j$ 
            envoie explorer( $z$ , prec) à  $k$ 

```

Question 8. Faire “tourner à la main” cet algorithme sur l’anneau dessiné ci-dessous (le site 0 est l’initiateur). Préciser les différentes étapes avec, à chaque fois, les affectations des variables successeur ou éléments du tableau R qui ont lieu, et pour quel site. Préciser également le message avec ses paramètres envoyé à chaque étape.



On réalise ensuite un tour de l'anneau au moyen de l'algorithme suivant (lancé par l'initiateur qui envoie `jeton(successeur)` à `R[premier]`)

Sur réception de `jeton(dest)` venant de `j`

Si `dest = monidentité`

alors

`<< utiliser la ressource donnée par le jeton >>`

`dest = successeur`

envoyer `jeton(dest)` à `R[j]`

Question 9. Là encore, faire tourner à la main cet algorithme en tenant compte des variables mises à jour dans la phase précédente, et mettre en évidence le fait que chaque site peut utiliser la ressource du jeton une et une seule fois au cours d'un parcours complet de l'anneau.