Département Licence	ANNÉE : 2005/2006 SESSION 1 **PARCOURS :** INF 6 **UE :** INF 108, Algorithmes des graphes **Épreuve :** INF 108 CC **Date :** Samedi 29 avril 2006 **Heure :** 8 heures 30 **Durée :** 1 heure 30 Documents : non autorisés

Nom : Prénom :

Avertissement

– Les cinq exercices sont indépendants.
– Réponses directement sur la feuille. (Utilisez un brouillon pour éviter les ratures sur la copie).
– L'ensemble est assez long, mais le total des points est de 25.

Le sujet

Exercice 1 (Dénombrements (2 points)
Soit F une forêt (graphe simple non orienté sans cycle) de n sommets et m arêtes.
Quel est le nombre de composantes connexes de F ? Justifiez votre réponse.

Exercice 2 (Chemins infinis (4 points)
Rappels : *Après un parcours en profondeur, un arc (u, v) d'un graphe orienté G est :*
– *un arc de liaison ou un arc avant si $d[u] < d[v] < f[v] < f[u]$*
– *un arc retour si $d[v] < d[u] < f[u] < f[v]$*
– *un arc couvrant si $d[v] < f[v] < d[u] < f[u]$*
Un chemin infini est une suite infinie d'arcs de G non nécessairement distincts (s_i, s_{i+1}).

Dans un graphe orienté fini $G = (S, U)$, montrez que les propositions suivantes sont équivalentes :

P1 : G possède un arc retour après un parcours en profondeur.

P2 : G possède un circuit.

P3 : Il existe un chemin infini dans G.

Exercice 3 (Parcours en profondeur (6 points))

Rappels : *Dans un graphe orienté* $G = (S, U)$

– *un chemin fini* $s \rightsquigarrow t$ *est une suite finie de* n *arcs* $(s_i, s_{i+1}) \in U$ *avec* $s = s_0$ *et* $t = s_n$.

– *un chemin élémentaire est un chemin fini tel que* $\forall(i, j), s_i \neq s_j$

– *un chemin élémentaire maximal à droite est un chemin élémentaire tel que* $\forall x \in S, s \rightsquigarrow t.(t, x)$ *n'est pas un chemin élémentaire.*

– *un chemin élémentaire maximal à gauche est un chemin élémentaire tel que* $\forall x \in S, (x, s).s \rightsquigarrow t$ *n'est pas un chemin élémentaire.*

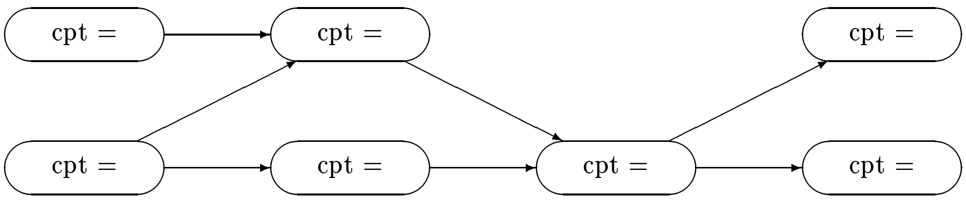
– *un chemin élémentaire maximal est un chemin élémentaire maximal à droite et à gauche.*

Soit la variante suivante du parcours en profondeur.

```
ParcoursComptage(G)
pour chaque sommet u de U[G]
     faire couleur[u] <- BLANC
           cpt[u] <- 0
pour chaque sommet u de U[G]
     faire si couleur[u] = BLANC
           alors Visiter_Comptage_PP(u)
```

```
Visiter_Comptage_PP(u)
couleur[u] <- GRIS
si Adj[u] = vide
alors cpt[u] <- 1
sinon pour chaque v de Adj[u]
           faire si couleur[v] = BLANC
                 alors Visiter_Comptage_PP(v)
                 cpt[u] <- cpt[u] + cpt[v]
couleur[u] <- NOIR
```

Question 3.1 (1 point) *Exécutez l'algorithme sur ce graphe.*



Question 3.2 (1,5 point) *Que fait cet algorithme lorsqu'on l'applique à un graphe orienté simple et sans circuit* (À quoi correspond la valeur `cpt[u]` calculée) *?*

Question 3.3 (1,5 point) *En supposant qu'il existe une structure Pred[u] qui donne pour chaque sommet la liste de ses prédécesseurs, donnez un algorithme qui lorsqu'on l'applique à un graphe orienté simple et sans circuit calcule le nombre de chemins élémentaires maximaux.*

Question 3.4 (2 points) *Quelle(s) modification(s) sont nécessaires pour traiter un graphe orienté sans arc multiple et sans circuit ?* (les boucles sur les sommets sont permises)

Exercice 4 (Comparaisons des parcours (6 points))

Question 4.1 (1 point) *Soit un arbre orienté $A = (S, U)$ et $r = racine(A)$. Montrez que les sous-graphes de liaison obtenus par les algorithmes* `ParcoursLargeur(A,racine(A))` *et* `ParcoursProfondeur(A,racine(A))` *sont identiques. (*`ParcoursProfondeur(G,s)` *est une variante qui consiste à ne visiter récursivement le graphe qu'à partir du sommet* `s`*).*

Question 4.2 (1 point) *Donnez un exemple de graphe simple orienté acyclique $DAG = (S, U, s \in S)$ tel que les sous-graphes de liaison obtenus par* `ParcoursLargeur(DAG,s)` *et* `ParcoursProfondeur(DAG,s)` *soient distincts.*

Question 4.3 (1 point) *Donnez un exemple de graphe simple orienté $G = (S, U, s \in S)$ tel que :*
– `ParcoursProfondeur(G,s)` *engendre au moins un arc avant,*
– *les sous-graphes de liaison obtenus par* `ParcoursLargeur(G,s)` *et* `ParcoursProfondeur(G,s)` *sont différents.*

Question 4.4 (1 point) *Donnez un exemple de graphe simple orienté $G = (S, U, s \in S)$ tel que :*
– `ParcoursProfondeur(G,s)` *engendre au moins un arc couvrant,*
– *les sous-graphes de liaison obtenus par* `ParcoursLargeur(G,s)` *et* `ParcoursProfondeur(G,s)` *sont différents.*

Question 4.5 (2 points) *Soit un graphe simple orienté $G = (S, U, s \in S)$.*
Montrez que si `ParcoursProfondeur(G,s)` *engendre aucun arc avant et aucun arc couvrant, les sous-graphes de liaison obtenus par* `ParcoursLargeur(G,s)` *et* `ParcoursProfondeur(G,s)` *sont identiques.*

Exercice 5 (Arbres couvrants (7 points))
On veut calculer X_n le nombre d'arbres couvrants de la grille $2 \times n$, c'est-à-dire le graphe
$G_n = (\{x_{i,j} | 1 \leq i \leq 2,\ 1 \leq j \leq n\}, \{\{x_{1,j}, x_{2,j}\} |\ 1 \leq j \leq n\} \cup \{\{x_{i,j}, x_{i,j+1}\} | 1 \leq i \leq 2,\ 1 \leq j < n\})$.
On note également $H_n = G_n - \{\{x_{1,n}, x_{2,n}\}\}$, et Y_n le nombre d'arbres couvrants de H_n.

FIG. *1 – La grille G_n et son sous-graphe H_n*

Question 5.1 (1 point) *Quels sont X_1 et Y_1 ?*

Question 5.2 (1 point) *Démontrez que, pour tout $n > 1$, $Y_n = X_{n-1}$*

Question 5.3 (1,5 point) *Soit* $n > 1$ *et* T *un arbre couvrant de la grille* $2 \times n$, *contenant les trois arêtes* $\{x_{1,n}, x_{2,n}\}, \{x_{1,n-1}, x_{1,n}\}, \{x_{2,n-1}, x_{2,n}\}$.
Montrez que T' *obtenu à partir de* T *privé de ces 3 arêtes, et augmenté de l' arête* $\{x_{1,n-1}, x_{2,n-1}\}$, *est un arbre couvrant de la grille* $2 \times (n-1)$.

Question 5.4 (1,5 point) *Exprimez, pour* $n > 1$, *le nombre* Z_n *d'arbres couvrants de la grille* $2 \times n$, *contenant les trois arêtes* $\{x_{1,n}, x_{2,n}\}, \{x_{1,n-1}, x_{1,n}\}, \{x_{2,n-1}, x_{2,n}\}$, *à partir des nombres* X_{n-1} *et* Y_{n-1}.

Question 5.5 (2 points) *Donnez une récurrence pour* X_n $(n > 1)$.