Parcours de graphes

Épreuve pratique d'algorithmique et de programmation
Concours commun des écoles normales supérieures
Durée de l'épreuve : 4 heures

Juillet 2004

Ce problème est consacré à l'étude et à l'adaptation d'algorithmes de parcours de graphes : le calcul des sommets accessibles et le calcul des composantes fortement connexes.

Le préambule définit les données qui seront appliquées aux algorithmes du problème. La partie 1 traite du problème de l'accessibilité pour plusieurs variantes de types de chemins et la partie 2 est consacrée au calcul des composantes fortement connexes d'un graphe. Il est **très fortement conseillé** de tester au préalable vos algorithmes sur des données plus petites.

Nous appelons complexité d'un algorithme, l'ordre de grandeur du nombre d'instructions exécutées suite au lancement de l'algorithme, par exemple $O(n)$, $O(n^2)$.

Attention, il est possible que certains calculs prennent trop de temps : dans ce cas, ne vous obstinez pas.

Préambule

Posons *NUMERO* égal à votre numéro inscrit sur votre table d'examem. Soit (x_i) la suite récurrente définie par $x_{i+1} = 3321 \cdot x_i + 5701$ et $x_0 = NUMERO$.

Question 1 *Donnez les valeurs* **modulo** 1000 *de* x_0, x_1, x_2, x_{20}, x_{100}, x_{1000}.

Un *graphe (orienté)* G est représenté par un couple (S, R), où S est un ensemble fini de sommets et R est une relation binaire sur S ($R \subseteq S \times S$). On dit que (u, v) est un arc de G si $R(u, v)$ (i.e, $(u, v) \in R$). Une suite de sommets $\sigma = \sigma_0, \ldots, \sigma_{n-1}$ est un *chemin* du graphe G si $R(\sigma_i, \sigma_{i+1})$ pour tout i ; le sommet σ_0 (resp. σ_{n-1}) est appelé *source* (resp. *destination*) du chemin σ. Nous dirons qu'un sommet v est accessible à partir d'un sommet u s'il existe un chemin dans G de source u et destination v. Le *degré sortant* d'un sommet est le nombre d'arcs qui en partent, et le *degré rentrant* d'un sommet est le nombre d'arcs qui y arrivent. Le degré sortant (resp., degré entrant) d'un graphe est le maximum des degrés sortants (resp., degrés entrants) des sommets.

Dans vos programmes, un graphe $G = (S, R)$ sera identifié par :
- $Q = \{0, \ldots, N-1\}$

– $R = \{(u, v) \in S \times S \mid \exists a \in \{0, \ldots, M-1\}, succ(u, a) = v\}$ où *succ* est une matrice $N \times M$ d'entiers.

Soient N et M deux entiers. Nous définissons le graphe $G_{N,M} = (\{0, \ldots, N-1\}, succ)$ où

$$succ(u, a) = x_{M*u+a} \mod N$$

Posons $\Gamma = \{G_{503,4}, G_{1007,8}, G_{5007,50}, G_{10003,40}, G_{49019,110}, G_{100003,150}\}$.

Question 2 *Quels sont les degrés sortants et les degrés entrants des sommets* $0, 1, 2, 3$ *et* 4 *pour les graphes de* Γ *? Notez que dans notre codage un arc peut être représenté plusieurs fois.*

Question 3 *Quels sont les degrés sortants et les degrés entrants des graphes de* Γ *?*

Question 4 *Combien d'arcs distincts contient chaque graphe de* Γ *?*

1 Problèmes d'accessibilité

Soit $G = (\{0, \ldots, N-1\}, R)$ un graphe. Posons $Acc(G)$ l'ensemble des sommets accessibles depuis le sommet 0. Notez que le calcul de $Acc(G)$ peut être simplement réalisé en initialisant $Acc(G)$ à $\{0\}$ et en lui ajoutant tout sommet v tel que $R(u, v)$ pour $u \in Acc(G)$; le calcul est terminé quand pour tout $u \in Acc(G)$ et pour tout $v \in Q$: $R(u, v) \Rightarrow v \in Acc(G)$.

Question 5 *Donnez une description concise de l'algorithme de calcul de l'ensemble* $Acc(G)$*. Donnez-en la complexité en fonction de la taille du graphe. Donnez le cardinal de l'ensemble* $Acc(G)$ *pour tous les graphes de* Γ.

Soit $\sigma = \sigma_0, \ldots, \sigma_{n-1}$ un chemin du graphe G. Posons

$$\max(\sigma, k) = \max_{i<k} \sigma_i$$

Le chemin σ est dit de type 1 avec une amplitude A si

$$\forall i \in \{0, \ldots, n-1\}, \sigma_i + A > \max(\sigma, i)$$

Posons $Acc_1(G, A)$ l'ensemble des sommets accessibles à partir du sommet 0 par un chemin de type 1 d'amplitude A.

Question 6 *Donnez une description concise de l'algorithme de calcul de l'ensemble* $Acc_1(G, A)$*. Donnez-en la complexité en fonction de la taille du graphe et de l'amplitude. Donnez le cardinal de l'ensemble* $Acc_1(G, A)$ *pour tous les graphes de* Γ *et pour les amplitudes* $A = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$ *et* 200*. Que remarquez-vous ?*

Question 7 *Posons* $Vacc_1(G)$ *l'ensemble des valeurs possibles de* $|Acc_1(G, A)|$ *pour* $A = 0, \ldots, N-1$*. Calculez* $Vacc_1(G)$ *(si* $|Vacc_1(G)| > 10$*, donnez seulement les 10 premières valeurs). Pour chaque* V *de* $Vacc_1(G)$*, donnez la plus petite amplitude* A *telle que* $|Acc_1(G, A)| = V$ *(si* $Vacc_1(G)$ *est trop grand, le calcul est restreint aux 10 premières valeurs).*

On dit qu'un chemin $\sigma = \sigma_0, \ldots, \sigma_{n-1}$ est de type 2 avec une amplitude A si

$$\forall \in \{0, \ldots, n-1\}, \sigma_i < \max(\sigma, i) + A$$

Posons $Acc_2(G, A)$ l'ensemble des sommets accessibles à partir du sommet 0 par un chemin de type 2 d'amplitude A.

Question 8 *Même question que la question 6 pour des chemins de type* 2.

Question 9 *Même question que la question 7 pour des chemins de type* 2.

On dit qu'un chemin σ est de type 3 avec une amplitude A si σ est de type 1 et 2 avec une amplitude A. Posons $Acc_3(G, A)$ l'ensemble des sommets accessibles à partir du sommet 0 par un chemin de type 3 d'amplitude A.

Question 10 *Même question que la question 6 pour des chemins de type* 3.

Question 11 *Même question que la question 7 pour des chemins de type* 3.

2 Problèmes de composantes fortement connexes

Soit $G = (S, R)$ un graphe. Soit D la relation d'équivalenve définie pour tout $u, v \in S$ par $D(u, v)$ si u est accessible depuis v et v est accessible depuis u. Les classes d'équivalence de la relation D sont appelées *composantes fortement connexes* du graphe G. Notons $CFC_G(s)$ (ou plus simplement $CFC(s)$) la composante contenant le sommet s.

Soient N et M deux entiers. Nous proposons de calculer les composantes fortement connexes incluses dans $Acc(G_{N,M})$. Cet algorithme est basé sur un parcours en profondeur du graphe. Quand une composante fortement est complètement parcourue, elle est traitée et retirée du graphe. Pendant l'exécution du parcours, nous distinguerons 3 états possibles d'un sommet : non visité, visité et éliminé. Cet algorithme utilise les types de données suivants :

- *chemin* est une liste $(u_0, a_0), \ldots, (u_k, a_k)$ représentant un chemin dans le graphe, i.e, tel que $succ(u_i, a_i) = u_{i+1}$ pour $i < k$.
- *etat* est un tableau de taille N associant à tout sommet son numéro de visite. Cependant $etat(u) = -1$ désigne un sommet non visité et $etat(u) = -2$ un sommet éliminé.
- *visite* est une liste d'entiers représentant la liste des sommets visités ordonnée par ordre de numéro de visite.
- *nbVisite* est un entier désignant le nombre de sommet dans l'état visité.
- *CFC* est une liste ordonnée de sommets contenant les numéros de visite d'un représentant des composantes fortement connexes dans le graphe restreint aux sommets visités.

Notre algorithme préserve à partir de ces données une connaissance complète des composantes fortement connexes : soit $CFC = v_0, \ldots, v_m$, pour $i < m$, $CFC(v_i)$ contient les sommets dont le numéro de visite est dans $[v_i, v_{i+1}[$ et pour $i = m$ ceux dont le numéro de visité est supérieur ou égal au numéro v_m.

L'initialisation de l'algorithme est donné par :

- $chemin = (0, 0)$,
- $etat = [0, -1, -1, \ldots, -1]$,
- $nbVisite = 1$,
- $visite = 0$,
- $CFC = 0$.

Notez que les données initiales permettent de retrouver l'unique composante $\{0\}$ dans le graphe réduit aux sommets visités (i.e., l'unique sommet 0). L'algorithme termine quand la liste *chemin* est vide. Supposons que la liste *chemin* est non vide. Soit (u_k, a_k) le dernier élément de la liste *chemin* et v_m le dernier de *CFC*. Alors l'algorithme progresse en respectant les règles suivantes :

1. si $a_k < M$ et $etat[succ(u_k, a_k)] = -1$ alors ajouter à la liste *chemin* $(succ(u_k, a_k), 0)$, ajouter $succ(u_k, a_k)$ à la liste *visite*, ajouter *nbVisite* à la liste *CFC*, réaliser $etat[succ(u_k, a_k)] = nbVisite$ et incrémenter *nbVisite*.
2. si $a_k < M$ et $etat[succ(u_k, a_k)] = -2$ alors incrémenter a_k,
3. si $a_k < M$ et $etat[succ(u_k, a_k)] >= 0$ alors retirer de *CFC* toutes les valeurs plus grandes strictement que $etat[succ(u_k, a_k)]$ et incrémenter a_k de 1,
4. si $a_k = M$ et $etat[u_k] = v_m$ alors $CFC(u_k)$ du graphe complet est composé des sommets dont le numéro de visite est plus grand que v_m. Ces sommets sont rangés à la fin de la liste *visite* à partir du sommet u_k. Après avoir réalisé un traitement demandé sur la composante fortement connexe (par exemple, compter le nombre de sommets), retirer ces sommets de la liste *visite*, et éliminer les sommets du graphe (i.e., $etat[u] = -2$ pour tout u de $CFC(u_k)$). Réaliser $nbVisite = v_k$, retirer v_m de la liste *CFC*, retirer (u_k, a_k) de la liste *chemin* et finalement si la liste *chemin* n'est toujours pas vide, incrémenter a_{k-1}.
5. si $a_k = M$ et $etat[u_k] \neq v_m$ alors retirer (u_k, a_k) de la liste *chemin* et si *chemin* n'est toujours pas vide, incrémenter a_{k-1}.

Question 12 *Réaliser un programme de calcul des composantes fortement connexes d'un graphe $G_{N,M}$. Ce programme devra renvoyer le nombre de composantes fortement connexes incluses dans $Acc(G_{N,M})$, ainsi que la taille de la plus grand composante.*

Nous cherchons à determiner combien de sommets au maximun peuvent être parcourus une infinité de fois par un chemin infini de source 0 dans un graphe $G_{N,M}$. La réponse est simple : soit le graphe $G_{N,M}$ ne contient pas de cycle et la réponse est 0; sinon la réponse est la taille de la plus grande composante fortement connexe. Maintenant, si nous imposons des conditions sur les chemins, la méthode nécessite d'être adaptée.

Question 13 *Posons $Infini_1(G, A)$ la valeur du maximun de sommets pouvant être parcourus par un chemin infini de type 1 d'amplitude A. Donnez les principes de l'algorithme de calcul de $Infini_1(G, A)$. Donnez le résultat de votre programme pour tous les graphes de Γ et pour les amplitudes $A = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$ et 200.*

Question 14 *Posons $VInfini_1(G)$ l'ensemble des valeurs possibles de $Infini(G, A)$ pour $A = 0, \ldots, N-1$. Calculez $VInfini_1(G)$ (si $|VInfini_1(G)| > 10$, donnez seulement les 10 premières valeurs). Pour chaque V de $VInfini_1(G)$, donnez la plus petite amplitude A telle que $Infini_1(G, A) = V$ (si $VInfini_1(G)$ est trop grand, le calcul est restreint aux 10 premières valeurs).*

Question 15 *Même question que la question 13 pour des chemins de type* 2.

Question 16 *Même question que la question 14 pour des chemins de type* 2.

Question 17 *Même question que la question 13 pour des chemins de type* 3.

Question 18 *Même question que la question 14 pour des chemins de type* 3.