Université de Marne-la-Vallée

Licence d'informatique et d'I.U.P.

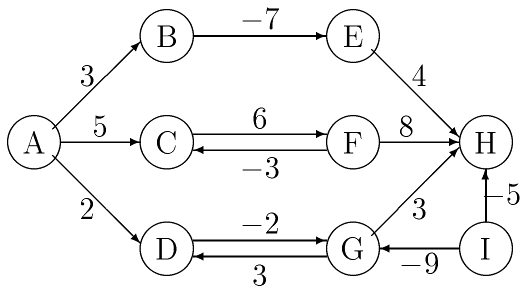
Examen d'algorithmique

Mai 2002, durée : 2h

Seule la consultation des supports de cours, travaux dirigés et travaux pratiques est autorisée pendant le contrôle.

Plus courts chemins

On considère le graphe G ci-dessous dont la source est le sommet A.



1. Décrire la forêt de l'exploration en profondeur du graphe G (sans les coûts des arcs) obtenue en considérant les nœuds dans l'ordre alphabétique à chaque fois que cela est nécessaire.

   Quel sont les arcs avant, les arcs arrière et les arcs de traverse de l'exploration ?

2. Appliquer l'algorithme de Bellman-Ford au graphe G et donner le résultat de chaque étape du calcul. On donnera la réponse sous la forme :

étape	d[A]	d[B]	d[C]	d[D]	d[E]	d[F]	d[G]	d[H]	d[I]
0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	...								
...	...								

../..

3. Montrer que l'on peut arêter le calcul de l'algorithme de Bellman-Ford si au cours d'une étape la table d n'est pas modifiée. Quelle est alors la conclusion du calcul et combien d'étapes sont effectuées par l'algorithme en fonction des arbres des forêts d'exploration ?
4. Pendant l'exécution de l'algorithme de Bellman-Ford, que peut-on conclure si un cycle est repéré dans le graphe des prédécesseurs ? Quelle est alors la conclusion du calcul ?
5. Pendant l'exécution de l'algorithme de Bellman-Ford, on dit qu'un sommet q du graphe est *actif* si les arcs partant de q n'ont pas été soumis à relaxation (c'est-à-dire, pas d'appel RELAX(q, r) pour les arcs de la forme (q, r)) depuis la dernière modification de $d[q]$. Montrer qu'à chaque étape il suffit de considérer les sommets actifs et les arcs qui en partent. Décrire des structures de données pour implanter cette stratégie.

Recherche d'un mot

Pour un mot $x \in A^+$ de longueur n ($x = x[0]x[1] \cdots x[n-1]$ et $x[0]$, $x[1]$, ..., $x[n-1] \in A$), on note $bord_x : \{0, 1, \ldots, n\} \longrightarrow \mathbf{N}$ la table définie par $bord_x[0] = -1$ et, pour $i = 1, 2, \ldots, n$, par

$$bord_x[i] = \max\{|u| : u \text{ préfixe et suffixe de } x[0]x[1] \ldots x[i-1], u \neq x[0]x[1] \ldots x[i-1]\}.$$

6. Calculer la table $bord_y$ pour $y =$ `aabaabaaabca`.

   Pour $x \in A^+$, indiquer comment trouver ses préfixes qui sont de la forme ww, $w \in A^+$, au moyen de la table $bord_x$? Quel est le temps de calcul de votre méthode ?
7. Écrire en langage C un algorithme qui teste si x est de la forme $ywwz$, où $y, z \in A^*$ et $w \in A^+$. Quel est son temps d'exécution ?