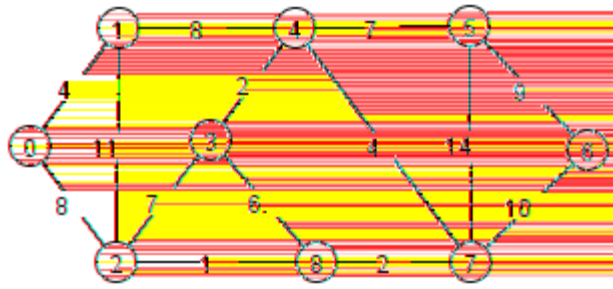


Algorithmique — M1
Examen du 18 janvier 2008
Université Paris Diderot

Documents autorisés : Deux feuilles de papier format A4
Durée : 3h

On applique les algorithmes de cours

Exercice 1 – Arbre couvrant minimum



Pour le graphe pondéré ci-dessus on cherche à trouver l'arbre couvrant minimum en appliquant un algorithme de cours.

1. Choisissez un algorithme (écrivez juste son nom s'il s'agit d'un algorithme connu).
2. Appliquez l'algorithme (dessinez toutes ses itérations).
3. Donnez le résultat final : arbre couvrant minimum.

Exercice 2 – Automate d'occurrences

1. Construire l'automate déterministe minimal qui reconnaît toutes les occurrences du mot "ARARAT" dans un texte quelconque. *Pour cet exercice seul le résultat final sera évalué. Inutile donc d'expliquer votre méthode.*

On adapte les algorithmes de cours

Exercice 3 – Poids max de camion

Un réseau routier connecte les villages d'une île. Pour chaque route $e = (u, v)$ on connaît le poids maximum (en tonnes) $p(u, v)$ du camion qui peut emprunter cette route. On cherche à calculer pour toutes les paires de villages u, v le poids maximum de camion qui peut aller de u à v via le réseau routier tout en respectant la contrainte de poids pour chaque route empruntée.

1. Formalisez le problème en complétant le texte ci-dessus.

On a un graphe (V, E) et une fonction de poids $p : E \rightarrow \mathbb{R}^+$. On définit la capacité d'un chemin $\sigma = v_1, v_2, \dots, v_k$ comme $C(\sigma) = \dots$. Pour deux sommets u et v on définit la capacité $C(u, v) = \dots$. Le problème algorithmique *CamionMax* consiste à trouver $C(u, v)$ pour toutes les paires de sommets u, v .

2. Quel algorithme de cours allez-vous modifier et quelles sont les modifications nécessaires ?
3. Donnez l'algorithme *CamionMax*(...)
4. Expliquez pourquoi votre algorithme est correct.
5. Analysez la complexité de votre algorithme.

Exercice 4 – Le jeu de doublets

Ce jeu attribué à Lewis Carroll consiste à relier deux mots donnés en une chaîne de mots différant d'une seule lettre de leurs voisins. Ainsi on va de CINQ à SEPT par la chaîne : CINQ, CINE, CENE, CENT, SENT, SEPT.

1. Représentez ce jeu comme recherche d'un chemin dans un graphe. Décrivez précisément le graphe.
2. L'approche naïve consiste à construire la totalité du graphe et à appliquer un des algorithmes de parcours vu en cours. Quelles sont les difficultés de cette approche ? (*une ou deux phrases svp*)
3. L'approche plus raisonnable consiste à appliquer un algorithme de parcours à la volée, c-à-d sans charger dans la mémoire la totalité du graphe. Supposons qu'on est fourni d'une fonction ("oracle") booléenne $\text{EstUnMot}(w)$ qui renvoie vrai si et seulement si w est un mot de la langue française. Écrivez un algorithme *Doublets*(u, v) qui cherche une chaîne de mots reliant u et v .

On invente un algorithme

Exercice 5 – La plus longue sous-séquence croissante

On a une séquence de m nombres différents $\alpha = a_1, a_2, \dots, a_m$. Le problème SSC consiste à trouver sa plus longue sous-séquence croissante et la longueur de cette sous-séquence.

Par exemple, pour 7, 1, 9, 2, 8, 4, 5, 3 la réponse est 1, 2, 4, 5, longueur 4.

On utilisera la programmation dynamique pour concevoir un algorithme qui résolve ce problème.

1. Soit $L(i)$ la longueur de la plus longue sous-séquence croissante dans i premiers éléments a_1, a_2, \dots, a_i de la séquence α . Essayez de trouver des équations de récurrence pour $L(i)$. Expliquez pourquoi c'est difficile.
2. Pour pallier à cette difficulté il faut prendre une fonction plus "fine" qui admette des équations de récurrence. On vous propose deux telles fonctions au choix :
 - $M(i)$ qui est la longueur de la plus longue sous-séquence croissante de α qui se termine par a_i (la sous-séquence doit donc inclure a_i).
 - $H(i, x)$ qui est la longueur de la plus longue sous-séquence croissante de a_1, a_2, \dots, a_i , telle que tous les éléments de cette sous-séquence sont inférieurs ou égaux à x .Choisissez une des deux fonctions H ou M (*ne faites pas les deux!*) Écrivez les équations de récurrence pour cette fonction sans oublier les cas de base.
3. Écrivez un algorithme efficace (récursif avec "marquage" ou itératif) pour calculer H ou M .
4. En sachant calculer la fonction choisie H ou M , comment répondre à la question initiale : trouver la longueur de la plus longue sous-séquence croissante de α .
5. Comment trouver la plus longue sous-séquence croissante elle-même (toujours en sachant calculer la fonction choisie H ou M).
6. Analysez la complexité de votre algorithme.