

Algorithmique — M1

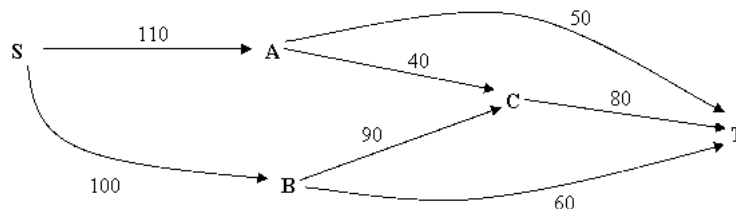
Examen du 11 janvier 2010

Université Paris Diderot

Documents autorisés : Deux feuilles de papier format A4
Durée : 3h (de 12 à 15)

On applique un algorithme de cours

Exercice 1 – Flux maximum



Pour le réseau ci-dessus on cherche à trouver le flux (flot) maximum 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 : flux maximum et sa valeur.

On adapte un algorithme de cours

Exercice 2 – 32 cavaliers

On cherche à disposer 32 cavaliers sur l'échiquier 8x8 pour qu'ils ne soient pas en prise (wikipedia dit que c'est possible). On cherche à développer un algorithme de type backtracking qui trouve une telle disposition. Par souci d'efficacité on souhaite voir chaque disposition une seule fois et non dans tous les ordres possibles.

1. Écrivez l'algorithme (en pseudocode)
2. Expliquez cet algorithme (vous pouvez vous inspirer de l'indication)
3. Estimez le nombre d'opérations nécessaire

Indication : Essayez de répondre aux questions suivantes pour parvenir à un tel algorithme.

- Comment tester que deux cavaliers ne sont pas en prise ?
- Comment représenter la disposition de 32 cavaliers par une structure de données ? Pour éviter une explosion inutile on propose de considérer les cavaliers de gauche à droite et de bas en haut.
- Comment définir une solution partielle ?
- Quelle est une solution partielle de taille 0 ?
- Comment augmenter une solution partielle ?
- Quand s'arrêter ?

On invente des algorithmes

Exercice 3 – La meilleure période

Le bénéfice net (positif ou négatif) de la société D&Q pendant n dernières années est représenté par le tableau $\beta = (b_1, b_2, \dots, b_n)$. Pour une période contiguë $i..j$ on définit le bénéfice cumulé $BC = (b_i + b_{i+1} + \dots + b_j)$. On s'intéresse à la meilleure période de l'histoire de la société où le bénéfice cumulé est maximal, et surtout à la valeur BCM de ce bénéfice cumulé maximal.

1. Pour $\beta = (-5, 1, 3, -1, 10, -2, 0)$ trouvez le BCM à la main.
2. Proposez un algorithme itératif simple ("naïf") qui calcule BCM. Estimez sa complexité.
3. Proposez un algorithme de type Diviser-Pour-Régner qui calcule BCM.

Indication.

- Soit la fonction $BCM(\beta, s, f)$ trouve la valeur de BCM à l'intérieur de la période $s..f$. On va programmer cette fonction.
 - Coupez le tableau en deux moitiés.
 - La meilleure période peut être dans la moitié gauche, dans la moitié droite ou bien à cheval entre les deux moitiés. Inventez comment traiter chacune de ces options, et comment en choisir la meilleure.
 - Donnez un algorithme récursif pour $BCM(\beta, s, f)$
4. Analysez la complexité de votre algorithme Diviser-Pour-Régner.

Exercice 4 – Les phrases

Une "phrase" est une séquence quelconque de mots français collés ensemble sans espaces ni ponctuation. Par exemple

gloutonalgorithmediviserpouregnerdynamique

est une phrase, par contre

abcdefbonjour

ne l'est pas.

On suppose dans cet exercice qu'une fonction booléenne $\text{mot}(w)$ est fournie. Elle renvoie vrai si la chaîne w est un mot français. Un appel de cette fonction prend une unité de temps.

Le problème algorithmique à résoudre dans cet exercice est le suivant : étant donné une chaîne $v = a_1 a_2 \dots a_n$ vérifier si c'est vraiment une phrase. On utilisera la programmation dynamique pour concevoir un algorithme polynomial qui résolve ce problème.

1. Soit $p(i)$ une fonction booléenne vraie si et seulement si le préfixe de v de longueur i (à savoir la sous-chaîne $a_1 a_2 \dots a_i$) est une phrase. Écrivez les équations de récurrence pour cette fonction sans oublier les cas de base.
2. Écrivez un algorithme efficace (récursif avec "marquage" ou itératif) pour calculer p .
3. En sachant calculer la fonction choisie p , comment répondre à la question initiale : est-ce que v est une phrase.
4. Analysez la complexité de votre algorithme.
5. Appliquez votre algorithme à la chaîne "bonjournaliste".
6. Comment modifier votre algorithme pour qu'il imprime à la fin la décomposition de la phrase en séquence de mots (si c'est possible).