L3-INFO Algorithmique et Modélisation

Quick – 24 mars 2023 – durée 1 h

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique).
Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés.
Il sera tenu compte de la qualité de la rédaction et de la clarté de la présentation (2 pts).
En cas d'incompréhension du sujet, préciser les hypothèses de travail que vous faites et continuer.
Le barème **indicatif** : Exercice 1 : 9 pts ($30 \sim mn$) ; Exercice 2 : 6 pts ($20 \sim mn$), Exercice 3 : 3 pts ($5 \sim mn$), Relecture : ($5 \sim mn$). Les 3 exercices sont indépendants.

I Dominos

On s'intéresse au problème de la fusion d'une suite de dominos, un domino fusionne avec un de ses deux voisins, de la façon suivante : les deux dominos voisins $\boxed{X \mid Y}$ $\boxed{Y \mid Z}$ se transforment en un unique domino $\boxed{X \mid Z}$ pour un coût de $X \times Y \times Z$.

EXEMPLE

Soit la suite $\boxed{4 \mid 5}$ $\boxed{5 \mid 8}$ $\boxed{8 \mid 5}$. Il y a deux possibilités pour la fusionner :

1. $\left(\boxed{4 \mid 5}\ \boxed{5 \mid 8}\right)\ \boxed{8 \mid 5}$

   Fusionner d'abord $\boxed{4 \mid 5}$ et $\boxed{5 \mid 8}$, qui donne $\boxed{4 \mid 8}$, pour un coût de $4 \times 5 \times 8$. Il reste à fusionner $\boxed{4 \mid 8}$ et $\boxed{8 \mid 5}$, ce qui coûte $4 \times 8 \times 5$.

2. $\boxed{4 \mid 5}\ \left(\boxed{5 \mid 8}\ \boxed{8 \mid 5}\right)$

   Fusionner d'abord $\boxed{5 \mid 8}$ et $\boxed{8 \mid 5}$, qui donne $\boxed{5 \mid 5}$, pour un coût de $5 \times 8 \times 5$. Il reste à fusionner $\boxed{4 \mid 5}$ et $\boxed{5 \mid 5}$, ce qui coute $4 \times 5 \times 5$.

Dans les deux cas, on obtient le domino $\boxed{4 \mid 5}$ et la première façon de faire coûte au total 320 et la seconde 300.

Voici une façon possible de mettre des parenthèses pour une suite de 6 dominos

$$\left(\boxed{5 \mid 3}\ \left(\left(\boxed{3 \mid 4}\ \boxed{4 \mid 6}\right)\ \boxed{6 \mid 2}\right)\right)\ \left(\boxed{2 \mid 2}\ \boxed{2 \mid 6}\right)$$

avec un coût de 222 (c'est à dire $3 \times 4 \times 6 + 3 \times 6 \times 2 + 5 \times 3 \times 2 + 2 \times 2 \times 6 + 5 \times 2 \times 6$)

On cherche le parenthésage qui permette de fusionner une suite de dominos avec un coût total minimal.

I.1. Voici une suite de 4 dominos : $\boxed{4 \mid 3}$ $\boxed{3 \mid 6}$ $\boxed{6 \mid 2}$ $\boxed{2 \mid 2}$

(a) Pour cet exemple, combien y-a-t-il de parenthésages possibles différents ?

(b) Pour chacun d'entre eux, donner le coût associé.

I.2. On s'intéresse à la meilleure façon de fusionner les dominos se trouvant entre le $i^{\text{ième}}$ et $j^{\text{ième}}$ domino (les dominos d'indices i et j étant inclus). On note $C(i, j)$ le coût de cette fusion. Établir une formule de récurrence pour les $C(i, j)$ avec $i < j$.

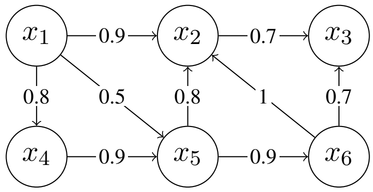
I.3. Proposer un algorithme efficace qui résolve le problème de fusion d'une suite de n dominos, et reposant sur un schéma de *programmation dynamique*. Évaluer le coût de cet algorithme

II Chemin le plus sûr

Un réseau de communication est constitué de nœuds et de liens (directionnels) permettant de relier les nœuds entre eux. Les liens sont susceptibles de tomber en panne, dans ce cas les liens ne peuvent pas acheminer de message. Pour chacun des liens, des mesures statistiques ont évalué la probabilité que le lien tombe en panne. L'objectif de cet exercice est de concevoir un algorithme qui détermine une route de fiabilité maximale permettant d'acheminer un message d'un nœud source à un nœud destination.

On modélise ce problème en représentant le réseau par graphe orienté $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$, avec $\mathcal{S} = \{x_1, \cdots, x_n\}$ l'ensemble des n sommets du graphe (modélisant les nœuds du réseau) ; $\mathcal{A}$ est l'ensemble des m arcs du graphe $\mathcal{G}$ (liens du réseau). Chaque arc (x_i, x_j) est étiqueté par une valeur $f_{i,j}$ qui représente la fiabilité de l'arc qui relie x_i à x_j. Une route est modélisée par un chemin qui relie un sommet source x_i à un sommet destination x_j.

Dans l'exemple suivant on a étiqueté les arcs par la fiabilité du lien. Ainsi la probabilité que l'arc (x_1, x_2) soit correct (c'est à dire ne soit pas en panne) est de 0.90.



II.1. Expliquer en une phrase pourquoi la fiabilité du chemin $[x_1, x_5, x_6]$ vaut 0.45. Existe-t'il un chemin de x_1 à x_6 ayant une meilleure fiabilité ? Si oui donner ce chemin et sa fiabilité.

II.2. Étant donné un graphe $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathcal{F})$. Proposer un algorithme efficace `Route_fiable` qui prend en argument 2 sommets et renvoie la valeur de la fiabilité maximale pour un chemin entre ces 2 sommets.

II.3. Modifier votre algorithme afin qu'il renvoie également un chemin de fiabilité maximale.

II.4. Démontrer la correction de votre algorithme (on pourra se ramener à des preuves d'algorithmes vus en cours).

III Un peu de complexité

On considère l'algorithme `magique` (T) qui renvoie une valeur magique à partir des valeurs des éléments du tableau T.

Fonction `magique` (T)
Données : T un tableau de taille n
Résultat : un entier

if *T est un tableau de 1 élément*
  renvoie `fonction_étrange` (cet élément)
else
  Diviser le tableau T en deux sous-tableaux (T_1 et T_2) de taille n/2
  Calculer $a_1 =$ `magique` (T_1) et $a_2 =$ `magique` (T_2)
  revoie `calcul_bizarre` (a_1, a_2)

Le coût par appel de `fonction_étrange` est c_e.
Le coût de division de T en deux sous-tableaux (T_1 et T_2) est 0.
Le coût par appel de `calcul_bizarre` est c_b.

III.1. Quel est le coût de cet algorithme, en fonction de a et b, lorsqu'on l'applique à un tableau de taille 8 ?

III.2. Soit le tableau T de taille $n = 2^k$ avec k un entier positif. Exprimer le coût de cet algorithme sur le tableau T en fonction de n, a et b. Expliquer brièvement les étapes du raisonnement.