CentraleSupélec

Algorithmique et complexité

TD 4/7 – Programmation dynamique

L'objectif de ce TD est de s'exercer à la modélisation de problèmes et à leur résolution par programmation dynamique.

Ce TD est en 3h et comprend des partie de pratique en Python.

Exercice 1 : Politique de placement

Nous disposons d'un euro que nous voulons placer d'une manière optimale pendant 10 périodes. Nous connaissons avec certitude le coefficient de gain c_i pour un placement en début de période i.

période i	1	2	3	4	5	6	7	8	9	10
c_i	2	4	1	2	6	2	2	4	1	4

A chaque période, on a deux options, soit on place la somme totale disponible, soit on garde toute la somme et on n'en place rien. Supposons qu'au début de la période i on dispose d'une somme x. S'il n'y a pas de placement en période i, la somme x sera disponible en début de période $i+1$. Si on place une somme x en début de période i on reçoit la valeur $c_i x$, mais l'argent sera immobilisé pendant cette période et pendant la période $i+1$ et la nouvelle valeur sera disponible seulement en début de période $i+2$ pour être réinvestie ou simplement récupérée.

Il faut trouver la politique de placement total qui rapportera le plus d'argent en début de la période 12 (ce qui signifie qu'il n'y aura plus de placement en début de la période 11).

Question 1

En ignorant dans un premier temps les coefficients de gains, modéliser le problème par un graphe non-pondéré (les sommets et les arcs) où chaque politique de placement totale correspondra à un chemin de ce graphe. De quelle nature est-il ? Dessiner le graphe.

Question 2

Si on veut se ramener à un problème du plus court chemin, quels poids faut-il affecter aux arcs ? Complétez le graphe.

Question 3

Quel algorithme vu en cours peut-on utiliser pour répondre à l'exercice ? Trouver la solution de l'exemple numérique.

Exercice 2 : Location de skis

Problème d'optimisation

Un magasin de ski possède en stock un ensemble de n paires de skis de différentes longueurs. Il reçoit une demande d'un club de vacances pour une location de m paires pour des clients dont la taille est connue avec $m < n$. Le propriétaire du magasin souhaite optimiser le confort des skieurs. Pour cela, il propose de minimiser la somme des écarts (en valeur absolue) entre la taille du client et la taille de ses skis.

Question 1

Formalisez ce problème d'optimisation.

Question 2

Considérons un algorithme glouton qui alloue les skis à l'aide de la méthode suivante : on cherche d'abord le couple (ski,client) qui a le plus petit écart *en valeur absolue*, on alloue la paire de ski au client et on recommence avec les skis et les clients restants.

Écrivez en Python le code de cet algorithme. Pour vous aider, rendez-vous sur la page de pratique du TD :

Question 3

Quelle est la complexité de cet algorithme ?

Question 4

L'algorithme glouton donne-t-il une solution optimale ?

Programmation dynamique

On suppose sans perte de généralité que les n paires de skis et les m skieurs sont classés par ordre croissant de taille. On note $Sol[i, j]$ le coût de la solution optimale (c'est-à-dire la somme des écarts de taille) pour les i premiers skis et les j premiers clients. On souhaite utiliser une méthode de type ≪ programmation dynamique ≫ pour trouver le coût de la solution optimale $Sol[n, m]$.

On peut constater que, pour 2 paires de skis et 2 skieurs, mieux vaut attribuer la plus petite paire au plus petit skieur et la plus grande paire au plus grand skieur. Ainsi, si on généralise ce résultat à i paires de skis et j skieurs, lorsque les $j - 1$ premiers skieurs sont servis, le skieur j peut choisir sa paire de ski et on ne trouvera pas de meilleure solution en permutant sa paire avec un skieur plus petit (servi avant lui).

C'est cette observation qui va nous permettre de définir notre algorithme de résolution par programmation dynamique.

Question 5

Donnez la formule de récurrence qui définit la valeur de $Sol[i, j]$ en fonction des valeurs de $Sol[i - 1, j]$ et $Sol[i - 1, j - 1]$.

Question 6

Retournez sur la page de pratique du TD pour écrire en Python un algorithme qui utilise la sous-structure optimale explicitée par la question précédente. Donnez la complexité de l'algorithme.

Flot Maximum – Coût minimum [Pour aller plus loin]

Les élèves les plus avancés, pourront explorer une seconde technique pour résoudre le problème de location de skis de manière exacte.

Le problème du flot maximum de coût minimum est le suivant. Étant données :

- un graphe orienté $G = (V, E)$;
- deux sommets source et puits $s \in V$, $t \in V$;
- une fonction de capacité $cap : E \to \mathbb{N}$;
- une fonction de coût $cost : E \to \mathbb{N}$;

⇒ trouver un flot maximum f : $E \to \mathbb{N}$, tel que $\sum_{e \in E} cost(e) * f(e)$ est minimum.

Question 7

Proposer une modélisation permettant de résoudre le problème d'affectation de skis en utilisant un problème de flot maximum de coût minimum.

Indice : On cherche un graphe avec des capacités unaires pour forcer le flot à être 0 ou 1.

Question 8

Pour coder en Python un algorithme de résolution d'un tel problème, rendez-vous de nouveau sur la page de pratique du TD.