

TD 1 : Programmation dynamique

Recherche opérationnelle S3.

2023

Exercice 1 — *L'argent de poche*

Un étudiant désire travailler en dehors de ses heures de cours pour gagner de l'argent de poche. Afin de ne pas compromettre ses études, il décide de consacrer un maximum de T heures par semaine à ses activités rémunératrices. Après de minutieuses recherches, il a trouvé n emplois possibles. Les salaires qui lui sont offerts ne sont pas proportionnels aux nombres d'heures de travail. Le tableau ci-dessous donne le salaire $s(j, i)$ de l'emploi i si on travaille j heures. T et n sont pris égaux à 4

heures de travail : x \ Emploi	Emploi I	Emploi II	Emploi III	Emploi IV
0	0	0	0	0
1	26	23	16	19
2	39	36	32	36
3	48	44	48	47
4	54	49	64	56

1. Pouvez-vous aider cet étudiant, en écrivant un algorithme naïf récursif, à déterminer quelle rémunération maximale il peut espérer ?
2. En déduire un algorithme de programmation dynamique, version mémoïsation, pour résoudre ce problème. Utilisez cet algorithme sur l'exemple.
3. Ecrire enfin la version itérative de l'algorithme de programmation dynamique. Utilisez cet algorithme sur l'exemple.
4. Quelles sont les complexités des 3 algorithmes ?
5. On souhaite, en plus de déterminer la rémunération optimale, savoir combien d'heure l'étudiant doit effectuer dans chaque emploi. Comment faire ?

Exercice 2 — *Distribution de scientifiques à des équipes de recherche*

Un projet de recherche spatiale doit résoudre un problème avant que des êtres humains ne puissent se rendre sur la planète Mars ... Quatre équipes de recherche indépendantes essayent des approches différentes dans cet objectif. On a estimé que leur probabilité d'échec est de 0.40, 0.60, 0.80, 1.0, respectivement. Ainsi la probabilité qu'aucune des équipes ne réussisse est de $0.40 \times 0.60 \times 0.80 \times 1.0 = 0.192$. Comme on cherche à minimiser la probabilité d'échec total, trois scientifiques de haut niveau ont été affectés au projet.

Le tableau suivant donne les probabilités d'échec de chaque équipe si 0, 1, 2 ou 3 scientifiques de plus lui sont affectés.

Nb scientifiques	Probabilité d'échec $P(i, j)$			
	Équipe			
	I	II	III	IV
0	0.40	0.60	0.80	1.00
1	0.20	0.40	0.50	0.90
2	0.15	0.20	0.30	0.10
3	0.10	0.19	0.20	0.10

1. Pouvez-vous aider à déterminer la probabilité d'échec minimale en écrivant un algorithme naïf récursif ?
2. En déduire un algorithme de programmation dynamique, version mémoïsation, pour résoudre ce problème. Utilisez cet algorithme sur l'exemple.

3. Ecrire enfin la version itérative de l'algorithme de programmation dynamique. Utilisez cet algorithme sur l'exemple.
4. Quelles sont les complexités des 3 algorithmes ?
5. On souhaite, en plus de déterminer la probability, savoir où affecter les scientifiques de haut niveau. Comment faire ?
6. L'équipe 2 a été retirée du projet. Peut-on déduire de la réponse à la question 2 la nouvelle probabilité d'échec du projet ? Sinon, expliquer succinctement comment procéder ?

Exercice 3 — Le problème de sac à dos

Ce problème fait partie des problèmes classiques de la Recherche Opérationnelle.

Etant donné un sac à dos de volume total V , n types d'objets, numérotés de 1 à n , avec m_i objets de type i , un volume v_i et une utilité u_i pour l'objet i .

on souhaite remplir le sac en maximisant l'utilité des objets qu'on y met.

On va utiliser la programmation dynamique pour résoudre ce problème. On définit x_i comme le nombre d'exemplaires de l'objet i mis dans le sac.

1. Donner l'utilité $U(x_1, x_2, \dots, x_n)$ du sac en fonction des objets qu'on a mis dedans.
2. Exprimer la contrainte de volume sous forme $V(x_1, x_2, \dots, x_n) \leq V$
On note $f(j, b)$ l'utilité maximale d'un sac de volume b rempli par les j premiers types d'objets uniquement, pour $b \in \llbracket 0, V \rrbracket$ et $j \in \llbracket 1; n \rrbracket$. C'est à dire la valeur maximale de $U(x_1, \dots, x_j)$ sachant que $V(x_1, \dots, x_j) \leq b$.
3. Si quelqu'un nous donne $f(j, b)$ pour tout j et tout b , comment, avec ces données, répondre au problème de sac à dos ?
4. Donner la valeur de $f(1, b)$ en discutant suivant les valeurs de b .
5. Donner une formule de récurrence permettant de calculer $f(j, b)$ pour $j \geq 2$ et tout b .
6. En déduire un algorithme naïf et un algorithme de programmation dynamique version mémoïsation pour ce problème. Quels est la complexité de ce dernier ?
7. Ecrire un algorithme de programmation dynamique itératif pour ce problème. Quelle est sa complexité ?
8. Appliquer l'un des deux algorithmes de programmation dynamique pour résoudre l'exemple suivant ayant 3 objets et un volume $V = 10$:

objet	1	2	3
v_i	3	5	2
m_i	2	2	4
u_i	10	15	8

Exercice 4 — Recherche de motif carré

Soit A une matrice $n \times m$ constituées de 1 et de 0. Décrire un algorithme permettant de trouver un carré de 1 de largeur maximum dans A , ainsi que les coordonnées du coin haut gauche de ce carré. Quelle est la complexité de cet algorithme ?

Exercice 5 — Monter des marches

1. Quel est le nombre de manières de monter m marches en les montant une par une et/ou deux par deux ?
Généralisons. Soit un escalier à m marches. On peut le gravir par différents sauts de α marches, pour $\alpha \in \{\alpha_1, \alpha_2, \dots, \alpha_p\}$. Par exemple, dans la question 1, on a $p = 2$, $\alpha_1 = 1$ et $\alpha_2 = 2$. On va supposer, par commodité, que les α_i sont rangés par ordre croissant et sont tous différents. On appelle $N(m)$ le nombre de manière de monter cet escalier.
2. Donner une formule de récurrence pour $N(m)$.
3. Donner l'algorithme de programmation dynamique associé (en pseudo code) et la complexité de cet algorithme.
4. Si $p = 3, \alpha_1 = 2, \alpha_2 = 3, \alpha_3 = 5$, décrivez $N(m)$ pour $0 \leq m \leq 12$.