Programmation dynamique

- Idée: Pour résoudre un problème, on commence par résoudre les plus petits sous-problèmes et on conserve les valeurs de ces sous-problèmes dans une table de programmation dynamique. On utilise ensuite ces valeurs pour calculer la valeur de sous-problèmes de plus en plus grands, jusqu'à obtenir la solution de notre problème global.

- C'est une approche du bas vers le haut

Retour monnaie

Problème: On a un montant M et des pièces de valeurs $v_1, v_2, \ldots, v_n$.
On veut trouver des entiers x_i tels que

$$\sum_{i=1}^{n} x_i v_i = M \quad \text{en minimisant} \quad \sum_{i=1}^{n} x_i$$

Solution: On va construire une table de programmation dynamique $C[1..n, 0..M]$.

où $C[i, j]$ = nombre minimum de pièces pour produire exactement le montant j en utilisant seulement des pièces de valeurs $v_1, v_2, \ldots, v_i$

La solution optimale sera en $C[n, M]$.

Retour monnaie

On a donc les équations suivantes:

$$C[i,0] = 0, \quad \forall i$$
$$C[i,j] = min(C[i-1,j], 1 + C[i, j-v_i])$$

Exemple: Supposons qu'on ait un montant de 8 et des pièces de valeurs $v_1 = 1, v_2 = 4$ et $v_3 = 6$:

	0	1	2	3	4	5	6	7	8
$v_1 = 1$	0	1	2	3	4	5	6	7	8
$v_2 = 4$	0	1	2	3	1	2	3	4	2
$v_3 = 6$	0	1	2	3	1	2	1	2	2

Programmation dynamique

- Question: Comment savoir si un problème d'optimisation peut être résolu par programmation dynamique?

Réponse:

- Principe d'optimalité: La solution optimale à un problème est composée de solutions optimales à des sous-problèmes

Problème du sac à dos

Problème: On dispose de *n* objets de poids positifs $w_1, w_2, \ldots, w_n$ et de valeurs positives $v_1, v_2, \ldots, v_n$. Notre sac à doc à une capacité maximale en poids de W

But: Maximiser $\sum_{i=1}^{n} x_i v_i$ tel que $\sum_{i=1}^{n} x_i w_i \leq W$ et $x_i \in \{0, 1\}$

On va construire une table de programmation dynamique $V[1..n, 0..W]$

où $V[i, j]$ = valeur maximale des objets que l'on peut transporter si le poids maximal permis est j et que les objets que l'on peut inclure sont ceux numérotés de 1 à i

La solution optimale sera en $V[n, W]$.

Problème du sac à dos

On a donc les équations suivantes:

$$V[i,0] = 0, \quad \forall i$$
$$V[i,j] = \max(V[i-1,j], V[i-1,j-w_i] + v_i)$$

Exemple: Supposons qu'on ait 5 objets de poids 1,2,5,6 et 7 et de valeurs $1, 6, 18, 22, 28$ et que la capacité de notre sac est de 11.

v_i	w_i		0	1	2	3	4	5	6	7	8	9	10	11
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
6	2	2	0	1	6	7	7	7	7	7	7	7	7	7
18	5	3	0	1	6	7	7	18	19	24	25	25	25	25
22	6	4	0	1	6	7	7	18	22	23	28	29	29	40
28	7	5	0	1	6	7	7	18	22	28	29	34	35	40

Plus court chemin

Problème: Soit G un graphe orienté, N l'ensemble de ces sommets et A l'ensemble de ces arêtes. Chaque arête à un poids positifs représentant la distance entre les 2 sommets. On veut calculer la longueur des plus courts chemins entre toutes les paires de sommets de G.

Solution: On va construire *n* matrices D_k pour $1 \leq k \leq n$.

où $D_k[i, j]$ = la longueur du plus court chemin entre *i* et *j* et dont les sommets intermédiaires sont dans l'ensemble $\{1, ..., k\}$

La longueur du plus court chemin entre chaque paire (*i*,*j*) de sommets est alors $D_n[i, j]$