

Principe de la programmation dynamique

1. La solution d'un problème P_k **contient** les solutions de problèmes ayant des tailles plus petites
2. La **construction** de la solution pour P_k s'effectue à l'aide d'un **faible nombre** d'informations sur P_j , $j < k$

Exemple 1: faire un produit de matrices de tailles différentes

On se donne n matrices M_1, M_2, \dots, M_n chaque M_i a m_i lignes et m_{i+1} colonnes, les entrées sont des nombres (réels ou entiers). On cherche la matrice produit:

$$M_1 M_2 \dots M_n$$

Il faut trouver l'ordre des produits à effectuer de façon à minimiser le nombre total d'opérations.

Remarque: Le calcul de $M_{i-1}M_i$ demande $O(m_{i-1}m_i m_{i+1})$ opérations.

Exemple Trois matrices M_1, M_2, M_3 de tailles respectives 10×100 , 100×5 et 5×50

1. Le calcul de M_1M_2 demande 5 000 multiplications de nombres le produit ensuite par M_3 fait faire 2 500 de plus soit un total de 7 500
2. Le calcul de M_2M_3 demande 25 000 multiplications de nombres, la détermination ensuite de $M_1M_2M_3$ utilise 50 000 multiplications soit un total de 75 000

Conclusion Il vaut mieux faire $((M_1M_2)M_3)$ que calculer $(M_1(M_2M_3))$

Cas général

- On note $c_{i,j}$ le nombre minimum de multiplications pour calculer $M_i M_{i+1} \dots M_j$
- On a alors $c_{i,i} = 0$, $c_{i-1,i} = m_{i-1} m_i m_{i+1}$
- On découpe $M_i M_{i+1} \dots M_j$ en $M_i \dots M_k$ et $M_{k+1} \dots M_j$ en choisissant le meilleur k :

$$c_{i,j} = \min_{k=i,j-1} [c_{i,k} + c_{k+1,j} + m_i m_{k+1} m_{j+1}]$$

- On doit déterminer $M_{1,n}$

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	-	-	-	-	-
2	-	0	-	-	-	-
3	-	-	0	-	-	-
4	-	-	-	0	-	-
5	-	-	-	-	0	-
6	-	-	-	-	-	0

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	15 750	-	-	-	-
2	-	0	2 625	-	-	-
3	-	-	0	750	-	-
4	-	-	-	0	1 000	-
5	-	-	-	-	0	5 000
6	-	-	-	-	-	0

Calcul de $c_{1,3}$:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$$c_{1,3} = \min [c_{1,1} + c_{2,3} + m_1 m_2 m_4, \quad c_{1,2} + c_{3,3} + m_1 m_3 m_4]$$

$$= \min [0 + 2625 + 30 \times 35 \times 5, \quad 15750 + 0 + 30 \times 15 \times 5]$$

$$= \min [7\ 875, \quad 18\ 000]$$

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	15 750	7 875	-	-	-
2	-	0	2 625	4 375	-	-
3	-	-	0	750	2 500	-
4	-	-	-	0	1 000	3 500
5	-	-	-	-	0	5 000
6	-	-	-	-	-	0

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	15 750	7 875	9 375	-	-
2	-	0	2 625	4 375	7 125	-
3	-	-	0	750	2 500	5 375
4	-	-	-	0	1 000	3 500
5	-	-	-	-	0	5 000
6	-	-	-	-	-	0

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	15 750	7 875	9 375	11 875	-
2	-	0	2 625	4 375	7 125	10 500
3	-	-	0	750	2 500	5 375
4	-	-	-	0	1 000	3 500
5	-	-	-	-	0	5 000
6	-	-	-	-	-	0

Un exemple:

i	1	2	3	4	5	6	7
m_i	30	35	15	5	10	20	25

$C_{i,j}$	1	2	3	4	5	6
1	0	15 750	7 875	9 375	11 875	15 125
2	-	0	2 625	4 375	7 125	10 500
3	-	-	0	750	2 500	5 375
4	-	-	-	0	1 000	3 500
5	-	-	-	-	0	5 000
6	-	-	-	-	-	0

Retrouver les étapes du calcul

un tableau pour les valeurs des k intermédiaires

$K_{i,j}$	1	2	3	4	5	6
1	-	-	1	3	3	3
2	-	-	-	3	3	3
3	-	-	-	-	3	3
4	-	-	-	-	-	5
5	-	-	-	-	-	-
6	-	-	-	-	-	-

Pour calculer $M_i \dots M_j$ on pose $k = K_{i,j}$ donné par le tableau puis on calcule $M_i \dots M_k$ et $M_{k+1} \dots M_j$ et on les multiplie entre elles.

Sac à dos

- Des objets $1, 2, \dots, n$ de poids p_1, p_2, \dots, p_n
- Chacun rapporte un bénéfice a_1, a_2, \dots, a_n
- Trouver le bénéfice maximum réalisable sachant que la charge maximale est P_0

Sac à dos: idée pour P_0 entier pas trop grand

- On note $B(k, p)$ le bénéfice maximal réalisable avec des objets $1, 2, \dots, k$ et le poids maximal p
- On a pour $k = 1$:

$$B(1, p) = \begin{cases} 0 & \text{si } p < p_1 \\ a_1 & \text{si } p \geq p_1 \end{cases}$$

- Pour $k > 1$

$$B(k, p) = \begin{cases} B(k-1, p) & \text{si } p < p_k \\ \max(B(k-1, p), B(k-1, p - w_k) + ba_k) & \text{si } p \geq w_k \end{cases}$$

Exemple pour le sac à dos

Objets	1	2	3	4	5	6	7	8
Poids	2	3	5	2	4	6	3	1
Bénéfices	5	8	14	6	13	17	10	4

Poids total $P_0 = 12$

Sac à dos:

Objets	1	2	3	4	5	6	7	8
Poids	2	3	5	2	4	6	3	1
Bénéfices	5	8	14	6	13	17	10	4

$B(k, p)$	0	1	2	3	4	5	6	7	8	9	10	11	12
$k = 1$	0	0	5	5	5	5	5	5	5	5	5	5	5

Sac à dos:

Objets	1	2	3	4	5	6	7	8
Poids	2	3	5	2	4	6	3	1
Bénéfices	5	8	14	6	13	17	10	4

$B(k, p)$	0	1	2	3	4	5	6	7	8	9	10	11	12
$k = 1$	0	0	5	5	5	5	5	5	5	5	5	5	5
$k = 2$	0	0	5	8	8	13	13	13	13	13	13	13	13

Sac à dos:

Objets	1	2	3	4	5	6	7	8
Poids	2	3	5	2	4	6	3	1
Bénéfices	5	8	14	6	13	17	10	4

$B(k, p)$	0	1	2	3	4	5	6	7	8	9	10	11	12
$k = 1$	0	0	5	5	5	5	5	5	5	5	5	5	5
$k = 2$	0	0	5	8	8	13	13	13	13	13	13	13	13
$k = 3$	0	0	5	8	8	14	14	19	22	22	27	27	27

Sac à dos:

$B(k, p)$	0	1	2	3	4	5	6	7	8	9	10	11	12
$k = 1$	0	0	5	5	5	5	5	5	5	5	5	5	5
$k = 2$	0	0	5	8	8	13	13	13	13	13	13	13	13
$k = 3$	0	0	5	8	8	14	14	19	22	22	27	27	27
$k = 4$	0	0	6	8	11	14	14	20	22	25	28	28	33
$k = 5$	0	0	6	8	13	14	19	21	24	27	28	33	35
$k = 6$	0	0	6	8	13	14	19	21	24	27	30	33	36
$k = 7$	0	0	6	10	13	16	19	23	24	29	31	34	37
$k = 8$	0	4	6	10	14	17	20	23	27	29	33	35	38