

Mějme na vstupu posloupnost k matic $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \dots, \mathbf{M}_k$, které chceme mezi sebou vynásobit v zadaném pořadí, tedy chceme získat matici \mathbf{M} , která vznikne jako

$$\mathbf{M} = \mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{M}_3 \cdot \dots \cdot \mathbf{M}_k$$

přičemž operátor \cdot chápejte jako standardní maticové násobení. Matice ve vstupní posloupnosti nejsou obecně čtvercové a mohou mít různé rozměry, avšak vždy jsou rozměry takové, že je násobení možné provést. Jak známo, násobení matic není komutativní, tedy $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$. Násobení matic je však asociativní, tedy $(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$. Pokud máme tedy například tři matice $\mathbf{A}, \mathbf{B}, \mathbf{C}$, a chceme získat výsledek násobení $\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}$, z hlediska výsledku nezáleží na tom, zda nejdříve vynásobíme $\mathbf{A} \cdot \mathbf{B}$ a toto vynásobíme zprava \mathbf{C} nebo vynásobíme nejdříve $\mathbf{B} \cdot \mathbf{C}$, a to celé zleva vynásobíme \mathbf{A} . Výsledek bude v obou případech stejný, avšak počet provedených operací násobení stejný obecně nebude. Naším cílem bude vymyslet algoritmus, který zjistí, v jakém pořadí je třeba provést násobení matic ve vstupní posloupnosti, abychom získali správný výsledek a zároveň počet provedených operací násobení byl nejmenší možný (jako operaci násobení v tomto případě uvažujte jakékoliv násobení dvou skalárních hodnot mezi sebou). Celý algoritmus by měl mít polynomiální složitost vzhledem k počtu matic ve vstupní posloupnosti.

Takový algoritmus je možné navrhnout s použitím principů dynamického programování, tedy tak, že problém rozdělíme na elementární podproblémy, z nichž budeme postupně skládat řešení větších a větších podproblémů, až dostaneme řešení celého původního problému.

Jelikož vymyslet takový algoritmus rozhodně není jednoduché, zadání je rozděleno do několika dílčích kroků, které řešte v pořadí, v jakém jsou zde uvedeny. Za správné vyřešení každého kroku zadání můžete dostat body.

Krok 1 – Máme-li dvě matice \mathbf{A} a \mathbf{B} , kde \mathbf{A} má m řádků a n sloupců a \mathbf{B} má n řádků a o sloupců, kolik operací násobení bude potřeba k provedení maticového součinu $\mathbf{A} \cdot \mathbf{B}$? (0.5 bodu)

Krok 2 – Máme-li dvě matice \mathbf{A} a \mathbf{B} , kde \mathbf{A} má m řádků a n sloupců a \mathbf{B} má n řádků a o sloupců, kolik řádků a kolik sloupců bude mít matice vzniklá vynásobením $\mathbf{A} \cdot \mathbf{B}$? (0.5 bodu)

Krok 3 – Jako elementární podproblém pro náš dynamický algoritmus můžeme uvažovat nalezení počtu potřebných operací pro vynásobení dvou matic mezi sebou. Označme $\mathbf{M}_{i,j}$, $i < j$, matici, která vznikne vynásobením všech matic od \mathbf{M}_i až po \mathbf{M}_j , tedy

$$\mathbf{M}_{i,j} = \mathbf{M}_i \cdot \mathbf{M}_{i+1} \cdot \mathbf{M}_{i+2} \cdot \dots \cdot \mathbf{M}_{j-1} \cdot \mathbf{M}_j$$

přičemž $\mathbf{M}_{i,i} = \mathbf{M}_i$, tedy například $\mathbf{M}_{1,1}$ je původní matice \mathbf{M}_1 , kterou máme zadanou a např. $\mathbf{M}_{1,3}$ je matice vzniklá jako $\mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{M}_3$. Po vyřešení kroků 1 a 2 můžeme snadno zjistit, kolik operací by bylo potřeba k vynásobení všech vedle sebe se nacházejících dvojic matic ve vstupní posloupnosti, tedy kolik operací bychom potřebovali k získání matic $\mathbf{M}_{1,2}, \mathbf{M}_{2,3}, \mathbf{M}_{3,4}, \dots, \mathbf{M}_{k-1,k}$

(za žádných okolností nebudeme matice skutečně násobit, pouze spočítáme, kolik operací bychom k tomuto násobení potřebovali a jaké rozměry by vynásobené matice měly). Nyní bychom chtěli postoupit o úroveň výše a nalézt počty operací (minimální) potřebné pro získání matic $M_{1,3}$, $M_{2,4}$, $M_{3,5}$, ..., $M_{k-2,k}$. Zde už je situace trochu složitější, protože každou z těchto matic můžeme získat dvěma způsoby:

$$M_{i,i+2} = M_i \cdot M_{i+1,i+2}$$

nebo

$$M_{i,i+2} = M_{i,i+1} \cdot M_{i+1,i+2}$$

(Tedy např matici $M_{1,3}$ můžeme získat buďto jako $M_1 \cdot M_{2,3}$ nebo $M_{1,2} \cdot M_3$.)

přičemž rozměry matic $M_{i,i+1}$ a $M_{i+1,i+2}$ a počty operací potřebné k jejich získání už známe z první úrovně algoritmu. Z těchto dvou možností samozřejmě chceme vybrat tu, kde celkový počet potřebných operací bude menší (opět nebudeme matice násobit, pouze chceme znát počet potřebných operací a rozměry výsledné matice). Nalezněte způsob (napište vzorec), jak pro obě možnosti zjistit, kolik operací bude potřeba k provedení daného násobení a to na základě znalosti rozměrů všech čtyř matic M_i , M_{i+2} , $M_{i,i+1}$, $M_{i+1,i+2}$ a počtu operací potřebných k získání matic $M_{i,i+1}$, $M_{i+1,i+2}$ (u matic M_i a M_{i+2} je to 0, protože je máme již na začátku a tudíž k jejich získání nejsou potřeba žádné operace) (1 bod).

Krok 4 – Zobecněte postup z kroku 3 tak, aby fungoval postupně i pro nalezení minimálních počtů operací potřebných k získání matic $M_{1,4}$, $M_{2,5}$, $M_{3,6}$, ..., $M_{k-3,k}$, dále pak matic $M_{1,5}$, $M_{2,6}$, $M_{3,7}$, ..., $M_{k-4,k}$ atd. až se dostanete k matici $M_{1,k}$. Pokuste se popsat slovy myšlenku tohoto postupu, případně použijte pseudokód nebo kód v nějakém programovacím jazyce (2 body)

Krok 5 – Uvažujte následující vstupní posloupnost matic

$$M_1, M_2, M_3, M_4$$

kde

M_1 má 10 řádků a 20 sloupců

M_2 má 20 řádků a 50 sloupců

M_3 má 50 řádků a 1 sloupec

M_4 má 1 řádek a 100 sloupců

a vyplňte pro tento vstup následující dvě tabulky, kde v druhé tabulce hodnota na pozici i,j odpovídá minimálnímu počtu operací potřebných k získání matice $M_{i,j}$, na pozici 1,4 by tedy měl být správný výsledek (1 bod)

Matice	Počet řádků	Počet sloupců	Minimální počet operací potřebný k získání této matice
M_1	10	20	0
M_2	20	50	0
M_3	50	1	0
M_4	1	100	0
$M_{1,2}$			
$M_{2,3}$			
$M_{3,4}$			
$M_{1,3}$			
$M_{2,4}$			
$M_{1,4}$			

	1	2	3	4
1	0			
2		0		
3			0	
4				0

Poznámka: berte v úvahu, že za **Krok i** dostanete nenulový počet bodů pouze tehdy, pokud budete mít správně vyřešený **Krok i – 1**. Dohromady lze získat až 5 bodů.