

Dinamik Programlama

Tanım = Böl ve Yönet'e bağlı olarak geliştirilmiştir

→ Böl ve Yönet yaklaşımı ortak alt programları

tekerrür tekerrür çözecektir.

Dinamik Programlama her alt problem:

bir kez çözer ve sonucu bir tabloda saklar.

→ Matris Zincir Çarpımı Formülü kullanılır

$$A_{i \times k} \times B_{k \times j} = \underline{\underline{C_{i \times j}}}$$

↓

elde etmek için

toplam çarpım

sonuç

$$i \times k \times j = \text{kod}$$

Çarpım Sıra

$$A_1: 10 \times 100$$

$$A_2: 100 \times 5$$

$$A_3: 5 \times 50$$

$$1-) (A_1 \cdot A_2) \cdot A_3 \rightarrow A_1 \cdot A_2 = \underline{10} \times 100 \times \underline{5} = 5000 \rightarrow \text{adet çarpım}$$

$$A_{12} = 10 \times 5$$

$$A_{12} \cdot A_3 = \underline{10} \times 5 \times \underline{50} = 2500 \rightarrow \text{adet çarpım}$$

$$A_{123} = \underline{10 \times 50} \rightarrow 7500 \text{ skaler çarpım}$$

matris boyutu

$$2-) A_1 \cdot (A_2 \cdot A_3) \rightarrow A_2 \cdot A_3 = 100 \times 5 \times 50 = 25000 \rightarrow \text{adet çarpım}$$

$$A_{23} = 100 \times 50$$

$$A_1 \cdot A_{23} = 10 \times 100 \times 50 = 50000 \rightarrow \text{adet çarpım}$$

$$A_{123} = \underline{10 \times 50} \rightarrow 7500 \text{ skaler çarpım}$$

matris boyutu

▽ Bu bir optimizasyon problemi.

○ Min sayıda çarpım yapılabilir

Çin matrisler hangi sıra ile çarpılmalı

Olas Pertezlene

$$(A_1, \dots, A_k) \cdot (A_{k+1}, \dots, A_n)$$

$$\sum_{\substack{\text{farkli} \\ \text{bitime}}}^b F(k) \cdot F(n-k)$$

↳ farkli bitime

$$F(n) = ?$$

$$F(n) = \frac{1}{n} \cdot \binom{2n-2}{n-1}$$

↳ Farkli serpin mecuttur

$$n=6 \text{ için}$$

$$\frac{1}{6} \cdot \binom{6}{3} = \frac{1}{6} \cdot \frac{6 \cdot 5 \cdot 4}{3 \cdot 2 \cdot 1} = 5$$

$$n=10 \text{ için}$$

$$\frac{1}{10} \cdot \binom{18}{9} = \frac{1}{10} \cdot \frac{2 \cdot 18 \cdot 17 \cdot 16 \cdot 15 \cdot 14 \cdot 13 \cdot 12 \cdot 11 \cdot 10}{8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}$$

= 4862 Farkli serpin mecuttur

$$n = 1 \dots$$

$$\begin{array}{cccccccccccc} 1, & 1, & 2, & 5, & 16, & 42, & 132, & 429, & 1430, & 4862, & 16796 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 = n \end{array}$$

Optimal Parentheses

$$\underline{A_{i \dots j}} = A_{i \dots k} \cdot A_{k+1 \dots j}$$

$\hookrightarrow A_i$ matrixe A_j matrixe better compare
 optimal parentheses der

$$1 \leq i \leq j \leq n$$

$$m[i, j] = A_{i \dots j} \text{ 's; hexplorat; in; gereten; min;}$$

Compare; sayun; iface; oder

$$\text{Problem} = A_{1 \dots n} : m[1, n]$$

$$A_{i \dots j} = A_{i \dots k} \cdot A_{k+1 \dots j}$$

Area Formula

$$m[i, j] = m[i, k] + m[k+1, j] + P_{i-1} P_k P_j$$

$$m[i, j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k < j} \{ m[i, k] + m[k+1, j] + p_{i-1} p_k p_j \} & \text{if } i < j \end{cases}$$

A_1, A_2, A_3 Seru

	1	2	3
3	7500 $k=2$	25000 $k=2$	0
2	5000 $k=1$	0	
1	0		

$A_1: 10 \times 100 (p_0 \times p_1)$

$A_2: 100 \times 5 (p_1 \times p_2)$

$A_3: 5 \times 50 (p_2 \times p_3)$

$$m_{i,j}^{k=1} = m_{i,i} + m_{j,j} + p_{i-1} p_k p_j$$

$\begin{matrix} p_0 & p_1 & p_2 \\ 10 & 100 & 5 \end{matrix}$

$$m_{i,j}^{k=2} = m_{i,k} + m_{k+1,j} + p_{i-1} p_k p_j$$

$\begin{matrix} p_1 & p_2 & p_3 \\ 100 & 5 & 50 \end{matrix}$

$$m_{i,j}^{k=1} = m_{i,i} + m_{j,j} + p_{i-1} p_k p_j$$

$\begin{matrix} p_0 & p_1 & p_3 \\ 10 & 100 & 50 \end{matrix}$

$$k=2 \rightarrow m_{i,j} = m_{i,k} + m_{k+1,j} + p_{i-1} p_k p_j$$

$\begin{matrix} p_0 & p_2 & p_3 \\ 10 & 5 & 50 \end{matrix}$

$$A_1 A_2 A_3 = (A_1 A_2) A_3$$

$\begin{matrix} \text{table} \\ k=2 \end{matrix}$

$k=2$ yzilmaz

	1	2	3	4	5	6
6	3	3	3	5	5	0
5	3	3	3	4	0	
4	3	3	3	0		
3	1	2	0			
2	1	0				
1	0					

i

j

$$s[i, j] = A_i A_{i+1} \dots A_j \text{ zincirini } A_k \text{ ve } A_{k+1}$$

arasında paranteza ayarlar & değeri

$$\begin{aligned}
 A_{1-6} &= (A_{1-3}) \cdot (A_{4-6}) \\
 &= (A_1 \cdot (A_2 \cdot A_3)) \cdot ((A_4 \cdot A_5) \cdot A_6) \rightarrow \text{en optimal çarpma seçi}
 \end{aligned}$$

Maliyet $\rightarrow O(n^2)$ çok alt problem dir.

Zaman $\rightarrow O(n^3)$ 'üzün hep ndiyen

Bellek $\rightarrow O(n^2)$ bellek ndiyen

