Definition

$TIME(t(n)) = \{L \mid L$ is a language decided by an $O(t(n))$ time TM. $\}$

Class of P

$$P = \bigcup_k TIME(n^k)$$

P - P contains all the problems that can be solved/decided in polynomial time.

For example Sorting $\in TIME(n \log n)$

$PATH \in TIME(n^2)$

Binary Search $\in TIME(\log n)$

Dynamic programming

– An efficient technique that can be used to solve problems that can be divided into smaller subproblems.

Matrix Multiplication

Given $M_1, M_2, \ldots M_n$, compute the product $M_1 M_2 M_3 \ldots M_n$, where M_i has dimensions $d_{i-1} \times d_i$ (or M_i containing d_{i-1} rows and d_i columns)

Objective: minimize the total scalar multiplications.

$$\underset{2\times 2}{\begin{pmatrix} a & b \\ c & d \end{pmatrix}} \underset{2\times 1}{\begin{pmatrix} e \\ f \end{pmatrix}} = \underset{2\times 1}{\begin{pmatrix} ae+df \\ ce+df \end{pmatrix}}$$

Ex: $A_{n\times n} \cdot B_{n\times n} \cdot X_{n\times 1} =$

$(A \cdot B)_{n\times n} \cdot X_{n\times 1} \Rightarrow = n^3 + n^2 = O(n^3)$

$A_{n\times n} \cdot (B \cdot X)_{n\times 1} \Rightarrow n\times n\times 1 + n\times n\times 1$

$= n^2 + n^2$

$= O(n^2)$

Rephrase problem:

Parenthesize the product $M_1 M_2 \cdots M_n$ in a way to minimize the # of scalar multiplication.

$M_1 M_2 M_3 \cdots p \cdots M_n$

Ex: $M_1 - 20\times 10$

$M_2 - 10\times 50$

$M_3 - 50\times 5$

$M_4 - 5\times 30$

$M_1 \times M_2 \times M_3 \times M_4$

$((M_1 M_2)(M_3 M_4))$ - 47500

$M_1 M_2 - 20\times 10\times 50 = 10060$

$M_3 M_4 - 50\times 5\times 30 = 7500$

$(M_1M_2)_{20\times50} \cdot (M_3M_4)_{50\times30} = 20\times50\times30$

$= 30000$

$= 43500$ (+)

$(M_1(M_2M_3)M_4)$

$M_2M_3 - 10\times50\times5 = 2500$

$M_1 \cdot (\quad)_{10\times5} - 20\times10\times5 = 1000$

$(\quad)_{20\times5} \cdot M_{4\,5\times30} - 20\times5\times30 = 3000$

+

6500

$$(M_1 M_2 M_3 \cdots M_k)(M_{k+1} \cdots M_n)$$

$P(n)$ = # of alternative parenthesizations of n matrices

$P(k)$ $P(n-k)$

$$P(n) = \begin{cases} 1 & \text{if } n = 1 \\ \sum_{k=1}^{n-1} P(k)\,P(n-k) & \text{if } n \geq 2 \end{cases}$$

$P(n)$ - n^{th} catalan number

$$P(n) = \frac{1}{n}\binom{2(n-1)}{n-1} \geq \frac{4^{n-1}}{2n^2-n} = \Omega\left(\frac{4^n}{n^2}\right)$$

$n = 10$; $P(n) \geq \Omega\left(\frac{4^{10}}{10^2}\right) \sim 10^4$

$n = 20$; $P(n) \geq \Omega\left(\frac{4^{20}}{20^2}\right) \sim 10^9$

Dynamic programming solution

$(M_1 M_2 M_3 \cdots M_k)(M_{k+1} \cdots M_n)$

- $m[i,j]$ is the number of multiplications performed using optimal parenthesization of

$M_i M_{i+1} \cdots M_j$

$M_i M_{i+1} \cdots M_k M_{k+1} \cdots M_j$

$d_{i-1} \times d_i$ $\quad$ $d_{k-1} \times d_k$ $\quad$ $d_k \times d_{k+1}$ $\quad$ $d_{j-1} \times d_j$

$$\begin{cases} m[i,j] = \min\limits_k \{ m[i,k] + m[k+1,j] + d_{i-1} \times d_k \times d_j \\ m[i,i] = 0 \end{cases}$$

$m[1,n]$ contains the optimal number of scalar multiplications for the product.

$M_1 - 20 \times 10$

$M_2 - 10 \times 50$

$M_3 - 50 \times 5$

$M_4 - 5 \times 30$

i \ j	1	2	3	4
1	0	10000	3500	
2		0	2500	
3			0	7500
4				0

$$m[1,2] = \min_k \left[m[1,k] + m[k+1,2] + \right.$$

$k=1$

$m[1,1] + m[2,2]$

$+ d_{i-1} \times d_k \times d_j$

$20 \times 10 \times 50$

$m[2,3] = m[2,2] + m[3,3] + d_{i-1} \times d_k \times d_j$

$10 \times 50 \times 5$

$= 2500$

$m[3,4] = m[3,3] + m[4,4] + d_{i-1} \times d_k \times d_j$

$50 \times 5 \times 30$

$m[1,3] = \min$

$k=1$

$m[1,1] + m[2,3] + d_{i-1} \times d_k \times d_j$

$0 + 2500 + 20 \times 10 \times 5$

$= 3500$

$k=2$

$m[1,2] + m[3,3] + d_{i-1} \times d_k \times d_j$

$10000 + 0 + 20 \times 50 \times 5$

$= 15000$