Problem: given a sequence of matrices $A_1 \cdots A_n$ to be multiplied, we want to minimize the # of multiplications.

To multiply 2 matrices A_{p*q} and B_{q*r}, their size must fit

$$\underset{p\times r}{C} = \overset{A_{p\times q}}{\begin{bmatrix} & \\ & \end{bmatrix}} * \overset{B_{q\times r}}{\begin{bmatrix} & \\ & \end{bmatrix}}$$

# of scalar multiplications is pqr to obtain C.

Q:/ Does it matter how we multiply matrices in the sequence w/ respect to the order?

A:/ yes.

e.g. $A_1 \quad A_2 \quad A_3$ we can multiply as

$\overset{①}{(A_1 A_2)} A_3$ or $A_1 \overset{②}{(A_2 A_3)}$

if A_1 is 10×100

A_2 is 100×5

A_3 is 5×50

then ① will involve $10 \times 100 \times 5 + 10 \times 5 \times 50 = 7500$

② will involve $10 * 100 * 50 + 100 * 5 * 50 = 75000$

Q:/ How to parenthesize to get the min # of multiplications?

If we exhaustively search all possible parenthesizations, there are more than 2^n cases $\rightarrow$ exhaustive brute force won't work!

Q:/ How DP can help?

The structure of an optimal parenthesization for the following

$$A_i A_{i+1} A_{i+2} \cdots A_j \rightarrow A_{i \ldots j}$$

What we can claim is that there must be a k $i \le k < j$ s.t. we will split the sequence in the optimal parenthesization

$$\underbrace{A_i A_{i+1} \cdots A_k}_{A_{i \ldots k}} \qquad \underbrace{A_{k+1} \cdots A_j}_{A_{k+1 \ldots j}}$$

We can state that the way we ( ) $A_{i \ldots k}$ in the optimal ( ) of $A_{i \ldots j}$ must be the optimal ( ) of $A_{i \ldots k}$.

(proof by contradiction)

Thus we can use the following idea

optimal $A_{i \ldots k} \longrightarrow$ optimal part of $A_{i \ldots k}$ in $A_{i \ldots j}$

Recursion: $\quad A_i \cdots A_j \quad 1 \le i \le j \le n$

$m[i,j]$: <u>min</u> # of scalar $*$ to compute $A_{i \ldots j}$

(we will look for $m[1,n]$)

suppose for each A_i, its dimension is $P_{i-1} \times P_i$

Q:/ What is $m[i,i]$?

A:/ 0 since there is no $*$ involved for a single matrix

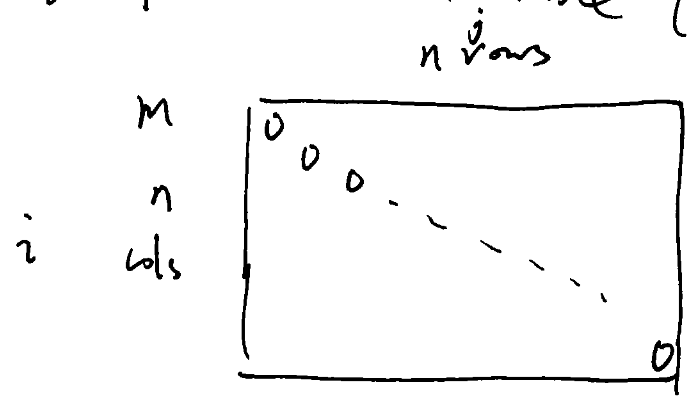
Thus

$$m[i,j] = \min \begin{cases} m[i,i+1] + m[i+2,j] + P_{i-1} P_{i+1} P_j \\ m[i,i+2] + m[i+3,j] + P_{i-1} P_{i+2} P_j \\ \vdots \end{cases}$$

$$= \min_{i \le k < j} \{ m(i,k] + m[k+1,j] + P_{i-1} P_k P_j \quad (\text{if } i<j)$$

So the recursion allows us to build up the optimal () of the problem of the optimal () of subproblems.

Size of m is n x n table (only the top half is used)



The table will be filled in diagonally.

	A_1	A_2	A_3	A_4	A_5	A_6
e.g.	30×35	35×15	15×5	5×10	10×20	20×25

	0	1	2	3	4	5	6
P	30	35	15	5	10	20	25

j

m	1	2	3	4	5	6
1	0	15750	7825	9375	11875	15125
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

i

our goal

fill in $m(i, i+1)$, then $m(i, i+2)$ ....

e.g. $m(1\ 2) = \min \begin{cases} m(1,1) + m[2\ 2] + P_0 P_1 P_2 \\ \end{cases}$

$= 30 \times 35 \times 15 = 15750$

$m[2,3] = \min \begin{cases} m[2,2] + m(3,3) + P_1 P_2 P_3 \\ \end{cases}$

$= 35 \times 15 \times 5 = 2625$

$$m(1,3) = \min \begin{cases} m(1,1] + m(2,3] + P_0 P_1 P_3 \\ m[1,2] + m(3,3] + P_0 P_2 P_3 \end{cases}$$

$$= \min \begin{cases} 0 + 2625 + 30 \times 35 \times 5 \\ 15750 + 0 + 30 \times 15 \times 5 \end{cases} = \min \begin{cases} 7875 \\ 18000 \end{cases} = 7825$$

The final sequence :

If we can trace back about how we obtained the optimal result, we can find the right ().

We record another matrix S s.t.

S[i,j] is k when we split $A_{i \ldots j}$ to get the optimal result.

S

i \ j	1	2	3	4	5	6
1	0	1	1	3	3	3
2		0	2	3	3	3
3			0	3	3	3
4				0	4	5
5					0	5
6						0

back trace from here

$$A_{1 \cdots 6} = (A_{1 \cdots 3}) * (A_{4 \cdots 6})$$

$$= ((A_1) * (A_{23})) * ((A_{4 \cdot 5})(A_6))$$

$$= ((A_1) * (A_2 A_3)) * ((A_4 A_5) * A_6)$$

The overall efficiency is $\Theta(n^3)$