Matrix Chain Multiplication

- A is an $n \times m$ matrix
- B is an $m \times k$ matrix
- How many scalar multiplications are needed to compute AB?

Matrix Chain Multiplication

- A is a 2×5 matrix
- B is a 5×1000 matrix
- C is a 1000×2 matrix.
- How many scalar multiplications are needed to compute ABC?

Matrix Chain Multiplication

- Input: A list of $n+1$ integers $p_1, p_2, \ldots, p_{n+1}$
- Output: The minimum number of scalar multiplications needed to compute $\Pi_{i=1\ldots n} A_i$ where A_i is a $p_i \times p_{i+1}$ matrix.
- Assume a standard matrix multiplication procedure is used; i.e. no Strassen-like improvements.

Inductive Approach cont.

- Consider an input: $p_1, p_2, p_3, p_4, p_5, p_6$
- Imagine an optimal solution: 10773

Inductive Approach cont.

- Consider an input: $p_1, p_2, p_3, p_4, p_5, p_6$
- Imagine an optimal way of multiplying matrices A_1, A_2, A_3, A_4, A_5:

$$(A_1(A_2A_3))\,(A_4A_5)$$

Inductive Approach cont.

- It prescribes some last multiplication

$$(A_1(A_2A_3)) \mid (A_4A_5)$$

Inductive Approach cont.

- It prescribes some last multiplication
  $(A_1(A_2A_3)) \mid (A_4A_5)$
- So $OPT(A_1,A_2,A_3,A_4,A_5) =$
  $OPT(A_1,A_2,A_3) + OPT(A_4,A_5) + p_1p_4p_5$

Inductive Approach cont.

- We don't know where the top split occurs … but clearly $OPT(A_1,A_2,A_3,A_4,A_5) =$

$$\min_{0<k<5} \; OPT(A_1,\ldots,A_k) + OPT(A_{k+1},\ldots,A_5) + p_1p_{k+1}p_5$$

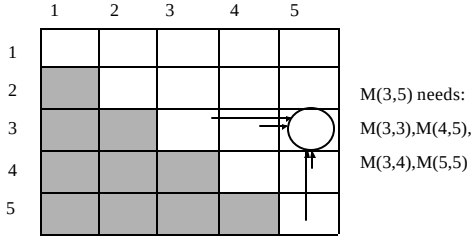
Recursive Algorithm Running Time

- $T(n) = \Sigma_{0<k<n} T(k) + T(n-k) + c$
- How bad is it?

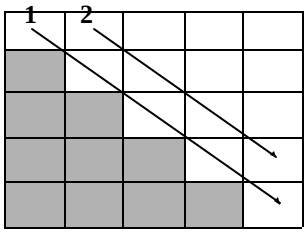
Dynamic Programming

- Use a table to store results
- What kind of results?
- M(k,j) = Minimum number of multiplications to compute $\Pi_{i=k \ldots j} A_i$

M(k,j) for k≤j



M(k,j) needs M(i,m) where m-i < j-k



M(k,j) needs M(i,m) where m-i < j-k

0				
	0			
		0		
			0	
				0

Dynamic Programming Algorithm

- M(k,k)=0
- For j,k such that j-k = 1, 2, …, n-1
  $M(k,j)= \min_{i=k \ldots j-1} M(k,i)+M(i+1,j) + p_k p_{i+1} p_j$
- Return M(1,n)

Input: 2,3,1,5,4,8 (A_1 is 2x3, A_2 is 3x1, …)

0	6			
	0	15		
		0	20	
			0	160
				0

MCM Algorithm

- Recursive Algorithm takes exponential time.
- Dynamic Programming takes _____.