# Optimal Parenthesization for Matrix Chain Product

Given a chain of N matrices $A_1, A_2, \cdots, A_N$, where for $i = \{1, 2, \cdots, N\}$ matrix $A_i$ has dimension $d_{i-l} \times d_i$, your task is to fully parenthesize the product $A_1 \times A_2 \times \cdots \times A_n$ in a way that minimizes the number of scalar multiplications required to evaluate this expression. Note that evaluating $A_1 \times A_2$ requires $d_0 \times d_1 \times d_2$ scalar multiplications.

For example, the matrix chain product $A_1 \times A_2 \times A_3$ can be parenthesized and evaluated in two possible ways:

1. $((A_1 \times A_2) \times A_3)$ requiring $d_0 \times d_1 \times d_2 + d_0 \times d_2 \times d_3$ scalar multiplications,
2. $(A_1 \times (A_2 \times A_3))$ requiring $d_0 \times d_1 \times d_3 + d_1 \times d_2 \times d_3$ scalar multiplications.

**Question 1**: Given a list $d$ which represents the chain of matrices such that matrix $A_i$ is of dimension $d[i-1]$ by $d[i]$, write a function, OPTPRODUCT, to solve this problem using a top-down dynamic programming approach. **Hint**: in each step, guess what should be outermost multiplication. Then write the base case and the recursive relation.

**Question 2**: Analyze the running time of your code.

**Question 3**: Write a function, OPTPRODUCTBOTTOMUP, to solve this problem using a bottom-up approach. **Hint**: find out the order in which you must solve the subproblems and determine what needs to be stored in a table.

**Question 4**: Analyze the running time of your code.