Description of Matrix Multiplier by Java.
Ali Akhavan
ID : 810180075

A Matrix Chain Multiplier is perhaps the quintessential example of dynamic programming, a technique that nearly every data structures and algorithms book explores. My implementation is no different from the rest, using *Introduction to Algorithms* by Cormen, Leiserson, and Rivest as the basis for its design. The problem can be stated as follows: given a chain $<A_1, A_2,..., A_n>$ of n matrices, where for i = 1, 2,...,n, matrix A_i has dimension p_{i-1} x p_i, fully parenthesize the product $A_1A_2 \cdots A_n$ in a way that minimizes the number of scalar multiplications. This program does exactly that, for a chain of up to 26 matrices. You may input your own dimensions for a matrix chain or have the computer generate them for you. If you choose to specify your own dimensions, you also have the option to input integer values for the matrix entries, since this program is capable of performing the multiplication.

It is imperative to understand how the input should look for this program to work properly. Use the following syntax:

matrix(*rows*, *cols*, [int, int,..., int]) * matrix(*rows*, *cols*, [int, int,..., int]) * ...

Rows and *cols* must, of course, be integers, and the number of entries within brackets must be exactly *rows * cols*.

Ex. (*for the matrix pictured above*): matrix(3, 3, [-362, -385, -408, 570, 675, 780, -362, -301, -240]) * matrix(...

If you choose not to input values for matrix entries, use the following syntax:
matrix(*rows*, *cols*) * matrix(*rows*, *cols*) * ...
Ex: matrix(3, 2) * matrix(2, 4)

Note: the *cols* of matrix A_{i-1} must equal the *rows* of Matrix A_i for all i.