**Dynamic Programming Definition**

“Therefore, any problem that can be expressed with a recursive mathematical formula can theoretically be solved using a recursive algorithm.”

“By rewriting the recursive algorithm as a non-recursive algorithm that systematically records the answers to the subproblems in a **table.**”

**A Leap of Faith**

In dynamic programming / recursive algorithms, the idea of “a leap of faith” suggest you assume any of your recursive call yields correct answer already. Therefore, this suggests a way of developing your recursive algorithms

1. Assume correct recursive sub-calls
2. Build your recursive sub-calls such that they are actually correct

**Recursive Mathematical Expressions**

Define operation C(N) such that

Find C(4).

**Optimal Ordering of Matrix Multiplication**

Doing the matrix multiplication in different order will result in drastically different number of computations made, therefore significantly impact your running time

A = (50 \* 10)

B = (10 \* 40)

C = (40 \* 30)

D = (30 \* 5)

Compute ABCD

(AB)(DC) requires 20000 + 6000 + 10000 = 36000 computations

A((BC)D) requires 12000 + 1500 + 2500 = 16000 computations

This demonstrates the LAW OF BEST ORDERING