**COT 5405 Homework #2**

**Due Monday Sept 28**

**Chained Matrix Multiplication:** We are given m matrices M1, M2, …, Mm where each successive pair is "compatible." That is, any successive pair can legally be multiplied. We assume Mi is dimension d(i) by d(i+1) (thus, Mi+1 is d(i+1) by d(i+2)) and all the d(i)'s are positive constants that are given to us – their exact values are not important for this homework. For example, the number of scalar multiplies to multiply M1\* M2 is d(1)\*d(2)\*d(3), and in general Mi \* Mi+1is d(i)\*d(i+1)\*d(i+2).

 That is, for any two successive matrices, we know the best (and only) way to multiply them. What about longer sequences?

 Let A(m, m) be an m by m array where a(i, j) is the minimum number of scalar multiplies needed to multiply the sub chain of matrices Mi \* Mi+1 \*… \* Mj. Notice that a(i, j) = a(j, i) – so half of the array values are kind of unimportant, and a(i, i) is sort of meaningless because it takes no multiplies to compute Mi, since it's given to you.

So, we may assume:

(1) d(1), d(2), …, d(m+1) is given to us,

(2) a(i, i) = 0, for 1 ≤ i ≤ m, and

(3) a(i, i+1) = d(i)\*d(i+1)\*d(i+2), for 1 ≤ i ≤ m–1.

Notice that the entry a(1, m) is the minimum number of scalar multiplies needed to multiply M1 \* M2 \*… \* Mm. (That is, the "answer" to the Chained Matrix Multiplication instance)

**Problem:** Suppose someone has filled all of the entries in the matrix A for you – except for the entry a(1, m) (and, of course, a(m, 1)).

1) Design an algorithm for determining the value of a(1, m).

 Remember, this should NOT be a program. It should be in relatively high level pseudo-code that includes a clear explanation of what is being done.

2) argue it's correctness.

3) derive it's complexity.