Due at the start of class Thursday, October 19, 2000.

Problem 1. Consider the following greedy algorithm for solving the chained matrix multiplication problem: Look for the two contiguous matrices that can be multiplied the fastest and multiply them. Continue like this until finished.

(More formally, let the dimensions for matrices $A_1, ..., A_n$, be given by the sequence $< p_0, p_1, \ldots, p_n >$. Look for the two contiguous matrices A_i and A_{i+1} whose multiplication minimizes the product $p_{i-1}p_ip_{i+1}$. Substitute p_{i-1}, p_{i+1} for p_{i-1}, p_i, p_{i+1} in p. Continue like this until p consists of only two values.)

Show that this greedy algorithm does not necessarily find the optimal way to multiply a chain of matrices.

Problem 2. Use the dynamic programming algorithm to find by hand an optimal parenthesization for multiplying matrices of dimensions are given by the sequence

$$< 6, 3, 10, 5, 8, 4, 20 > \quad .$$

Show the table. You may use a calculater.

Problem 3. Write an efficient algorithm to determine an order of evaluating the matrix product $M_1 \times M_2 \times M_3 \ldots \times M_n$ so as to minimize the scalar multiplications in the case where each M is of dimension $1 \times 1, 1 \times d, d \times 1, d \times d$ for some fixed d.

Problem 4. The number of combinations of n things taken m at a time $\binom{n}{m}$ can be computed using the following recurrence:

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1} \qquad \text{for } 0 < m < n$$

and

$$\binom{n}{0} = \binom{n}{n} = 1 .$$

(a) Write a recursive algorithm to compute $\binom{n}{m}$ using the above recurrence.

(b) Analyze the worst-case running time of your algorithm as a function of m and n.

(c) Produce a *memoized* version of your algorithm (see CLR pp. 312-314).

(d) Give a dynamic programming algorithm to compute $\binom{n}{m}$.

(e) Analyze the running time of your dynamic programming algorithm as a function of m and n.

Problem 5. Do Problem 16-2 on page 325 of CLR.