1. Placing Parentheses

   You are given an arithmetic expression containing n real numbers and $n-1$ operators, each either $+$ or x. Your goal is to perform the operations in an order that maximizes the value of the expression. That is, insert parentheses into the expression so that its value is maximized.

   For example:

   - For the expression $6*3+2*5$, the optimal ordering is to add the middle numbers first, then perform the multiplications: $(6*(3+2))*5 = 150$.
   - For the expression $0.1*0.1+0.1$, the optimal ordering is to perform the multiplication first, then the addition: $(0.1*0.1)+0.1 = 0.11$.
   - For the expression $(-3)*3+3$, the optimal ordering is $((-3)*3)+3) = -6$.

   (a) **(10 points)** Clearly state the set of subproblems that you will use to solve this problem.

   Solution:

   This problem is *very* similar to the matrix chain multiplication problem done in lecture.

   First, we define some notation. Denote the numbers by $a_1, a_2, \ldots, a_n$, and the operators by $op_1, op_2, \ldots, op_{n-1}$, so the given expression is $a_1\ op_1\ \ldots\ op_{n-1}\ a_n$.

   Let $M[i,j]$ be the *maximum* value obtainable from the subexpression beginning at a_i and ending at a_j (i.e., $a_i\ op_i\ \ldots\ op_{j-1}\ a_j$), and let $m[i,j]$ be the *minimum* value obtainable from the subexpression beginning at a_i and ending at a_j.

   We must keep track of both the minimum and the maximum because the maximal value of an expression may result from multiplying two negative subexpressions.

   (b) **(10 points)** Write a recurrence relating the solution of a general subproblem to solutions of smaller subproblems.

   Solution:

   To solve the subexpression $a_a \ldots a_b$, we can split it into two problems at the kth operator, and recursively solve the subexpressions $a_a \ldots a_k$ and $a_{k+1} \ldots a_b$. In doing so, we must consider all combinations of the minimizing and maximizing subproblems.

   The base cases are $M[i,i] = m[i,i] = a_i$, for all i.

$$\begin{aligned} M[a,b] = \max_{a \le k < b}(\max(&M[a,k]\ op_k\ M[k+1,b], \\ &M[a,k]\ op_k\ m[k+1,b], \\ &m[a,k]\ op_k\ M[k+1,b], \\ &m[a,k]\ op_k\ m[k+1,b])) \end{aligned}$$

$$\begin{aligned} m[a,b] = \min_{a \le k < b}(\min(&M[a,k]\ op_k\ M[k+1,b], \\ &M[a,k]\ op_k\ m[k+1,b], \\ &m[a,k]\ op_k\ M[k+1,b], \\ &m[a,k]\ op_k\ m[k+1,b])) \end{aligned}$$

(c) **(5 points)** Analyze the running time of your algorithm, including the number of subproblems and the time spent per subproblem.

Solution:

There are $O(n^2)$ subproblems, two for each pair of indices $1 \le a \le b \le n$. The subproblem $M[a,b]$ must consider $O(b-a) = O(n)$ smaller subproblems. Thus the total running time is $O(n^3)$.

2. Jewel Breaking

   Suppose you have a circular necklace with n jewels, each with some value, v_i. You wish to sell the jewels individually, but unfortunately, removing jewel i from the necklace breaks the neighboring jewels, making them worthless (i.e., $v_{i-1} = v_{i+1} = 0$). Formulate a DP problem to figure out the maximum revenue you can recieve from the circular necklace.

   Note: Notice that something special happens to the necklace after you decide whether or not to sell the first jewel. The resulting problem becomes a standard and simple DP formulation.