1. exercise R-5.9, p 282 [6 points]

2. For this dynamic programming problem and the next one, be sure to

(a) describe the subproblem
(b) give a recurrence for the subproblem
(c) provide pseudo-code showing how a table for the subproblems is filled
(d) give the time and space requirements of your method

Suppose we have two transmitters, each of which sends out repetitions of some short string. For example, transmitter 1 may repeat string \( x = 101 \) over and over, so what we will hear from it will be a prefix of \( x^k \) - that is, \( x \) concatenated to itself \( k \) times, possibly with a few bits chopped of the end (as in 10110110). Transmitter 2 repeats another string, \( y \). Our job is to determine if a sequence \( s \) that we have heard is an interleaving of these two transmissions.

For example, suppose transmitter 1 repeats \( x = 101 \) and transmitter 2 repeats \( y = 01 \). The sequence 010111010101 can be unraveled into \( x \) and \( y \): positions 1, 5, 9, and 12 contain 0101, a repetition of \( y \), while the remainder of the string contains 10110110, a repetition of \( x \).

Describe an efficient algorithm which takes a sequence \( s \) of length \( n \), and two strings \( x \) and \( y \), and determines if \( s \) is an interleaving of repetitions of \( x \) and \( y \). [8 points]

3. (exercise 6.6, DPV) We define a multiplication operation table on three symbols \( a, b, c \) according to something like the table below, so that \( ab = b \), \( ca = a \), and so on. The goal is to determine whether a string of symbols, \( s_1s_2 \cdots s_n \) (\( s_i \in \{a, b, c\} \)) can be parenthesized in such a way so that their multiplication together equals \( a \). Note that the operation is neither associative nor commutative (that is, it is possible that \( ab \neq ba \) and \( (ab)c \neq a(bc) \)).

Table 1: multiplication table for a, b, c

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>a</td>
<td>c</td>
<td>c</td>
</tr>
</tbody>
</table>

For example, if the input string is \( bbbbac \) then the answer should be **true** since \( ((b(bb))(ba))c = a \). 

*Hint:* define a subproblem \( P[i, j, t] \), where \( 1 \leq i \leq j \leq n \) and \( t \in \{a, b, c\} \). \( P[i, j, t] \) will be **true** if it is possible to parenthesize \( s_is_{i+1} \cdots s_j \) in such a way that the multiplication in that order equals the symbol \( t \), and **false** otherwise.

[8 points]
4. For the previous problem, show what needs to be saved in order to print out the way to parenthesize $s$. Give pseudo-code to do so. [5 points]

5. exercise 6.26 from DPV. Just give subproblem and recurrence. This should be a small modification of the Edit Distance problem (or LCS). [5 points]

Total: 32 points