This exercise sheet is concerned with dynamic programming. If you have no experience, it can be very hard to apply dynamic programming directly. It may help to first design a recursive solution for understanding the problem, and then transform it into a dynamic program. An example of this transformation can be found in the sample solution for programming part 1.

A complete description of a dynamic program always consists of the following aspects (relevant for the exam!):

1. **Definition of the DP Table**: What are the dimensions of the table? What is the meaning of each entry?
2. **Computation of an Entry**: How can an entry be computed from the values of other entries? Which entries do not depend on others?
3. **Calculation order**: In which order can entries be computed so that values needed for each entry have been determined in previous steps?
4. **Extracting the solution**: How can the final solution be extracted once the table has been filled?

The running time of a dynamic program is usually easy to calculate by multiplying the size of the table with the time required to compute each entry. Sometimes, however, the time to extract the solution dominates the time to compute the entries.

**Exercise 7.1 Numerical Puzzle.**

You are given a sequence of \( n \) digits 0,\ldots,9, and a positive integer \( \sigma \). Various sums can be obtained by inserting plus signs in different positions between the digits. If digits are not separated by plus signs, they are treated as a single decimal number.

**Example**: For the sequence \([6 \ 9 \ 2 \ 5 \ 0 \ 2 \ 1 \ 3]\), we can obtain the sums \( 69 + 2 + 5 + 0 + 21 + 3 = 100 \) and \( 6 + 9 + 250 + 21 + 3 = 289 \).

The exercise is to decide whether plus signs can be inserted into a given sequence, such that the sum equals \( \sigma \).

(a) Design an efficient algorithm for this problem using dynamic programming. You may assume that \( \sigma \) is relatively small in relation to \( n \).

**Hint**: Note that you only need to decide whether \( \sigma \) can be achieved or not.

(b) What is the asymptotic running time of the algorithm in b). Is it polynomial in the size of the input?

c) How can we efficiently find all arrangements of plus signs that yield the desired sum?
Exercise 7.2  Ascending Sequences (from an exam in February 2010).

In this exercise, we consider a two-dimensional array $A$ with $n$ rows and $m$ columns. The element $A[i][j]$ is adjacent to the elements $A[i-1][j], A[i][j-1], A[i+1][j]$ and $A[i][j+1]$, if these elements exist (elements at the borders of the array are adjacent to correspondingly fewer elements).

A sequence $x_1, x_2, \ldots, x_k$ of elements in the array is called ascending sequence if it satisfies the following conditions:

- the elements in the sequence are sorted in ascending order, and
- for every $i \in 1, \ldots, k - 1$, the elements $x_i$ and $x_{i+1}$ are adjacent in the array.

a) Design an efficient algorithm that finds the longest ascending sequence in a given array, that lies completely in one row or in one column. In the example below, a possible sequence would be 4, 6, 10. But this is not the longest sequence: the sequence 6, 28, 29, 47 is longer. Describe your algorithm, and specify its running time.

b) We now consider a longest ascending sequence in a given two-dimensional array. In the example below, a possible sequence would be 4, 6, 28, 29, 47, 49. Design the most efficient algorithm that finds such longest ascending sequence using dynamic programming. Describe the algorithm, and specify its running time.

Example-Array:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>27</td>
<td>42</td>
<td>41</td>
<td>48</td>
</tr>
<tr>
<td>35</td>
<td>39</td>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>49</td>
<td>2</td>
<td>38</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>47</td>
<td>29</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>25</td>
<td>33</td>
<td>10</td>
</tr>
</tbody>
</table>

Hand-in: until Wednesday 18th April 2012.