This exercise sheet is the third of three sheets that allow you to earn bonus points for the exam. In particular, you earn 2% of the total exam score as bonus points if you achieve at least 10 points from the following exercises. Please do not forget to justify your answers, when requested.

7 P Exercise 9.1 Mars mission.

The rover Curiosity landed on Mars and is located at a starting position \( S \). The goal is to move to a target position \( Z \), and to collect rock samples that are as valuable as possible. To not use too much energy, the rover is only allowed to take a step to the east (right) and to the south (down). The value of each rock sample is stored in an \( m \times n \) matrix, e.g.

\[
\begin{array}{ccccccc}
S & 9 & 2 & 5 & 11 & 8 & \\
& 17 & 21 & 32 & 5 & 15 & 3 \\
& 2 & 2 & 3 & 8 & 1 & 5 \\
& 8 & 2 & 8 & 11 & 15 & 9 \\
& 0 & 5 & 3 & 10 & 4 & Z \\
\end{array}
\]

In the matrix above, an example of a south-east path from \( S \) to \( Z \) is shown where the value of the collected rock samples is maximal. This path can be described by enumerating the corresponding matrix positions: \((1,1) \rightarrow (2,1) \rightarrow (2,2) \rightarrow (2,3) \rightarrow (2,4) \rightarrow \cdots \rightarrow (5,6)\).

a) Provide a dynamic programming algorithm that takes an \( m \times n \) matrix \( A \) with \( A[1,1] = A[m,n] = 0 \), and computes a south-east path from \( S = (1,1) \) to \( Z = (m,n) \) where the value of the collected rock samples is maximal. Note that we search for the path itself, and not just for the maximum value. To describe the algorithm, follow the scheme presented on exercise sheet 7.

b) Provide the running time of your solution in dependency of \( m \) and \( n \).

4 P Exercise 9.2 Depth-First Search, Breadth-First Search, Topological Sorting.

Consider the following graph \( G = (V,E) \).

![Graph](image)

a) Perform a depth-first search on \( G \) starting from the vertex \( A \). If more than one possible successor exists, use the lexicographic smallest vertex.
b) Perform a breadth-first search on \( G \) starting from the vertex \( A \). Use again the lexicographic smallest vertex as successor.

c) Provide a set \( E' \subseteq E \) of edges such that \( G' = (V, E \setminus E') \) can be sorted topologically. Provide a topological order of \( G' \).

**2 P Exercise 9.3 Minimum Spanning Tree.**

Use Kruskal’s algorithm to compute a minimum spanning tree for the graph below. Mark the edges contained in your solution.

![Graph](image)

**5 P Exercise 9.4 Black Holes.**

Let \( G = (V, E) \) be a directed graph. A black hole is a vertex \( v \in V \) with an indegree of \( |V| - 1 \) and an outdegree of 0. Describe an algorithm that gets the adjacency matrix of the graph \( G = (V, E) \) as input, and that checks whether \( G \) contains a black hole or not by considering only \( O(|V|) \) many matrix entries.

*Note:* Of course, a black hole can be found in time \( \Theta(|V|^2) \) by considering all entries of the adjacency matrix. We search for a more efficient solution.

**2 P Exercise 9.5 Union-Find Structures.**

In union-find structures, sets are represented by trees. We consider the process of “unification by size” (see Chapter 6.2.2). We want to create a tree of height \( h \in \mathbb{N} \). Describe how such a tree is generated with a sequence of UNION operations. How many UNION operations are necessary at least, and how many nodes does the resulting tree have?

**Hand-in:** until Wednesday, 8th May 2013.