Course 02249, DTU Compute
Carsten Witt

Computationally Hard Problems – Fall 2015
Assignment Project

Date: 06.10.2015, **Due date**: 02.11.2015, 23:59

This project counts for three weekly assignments. It should be performed in groups consisting of either two or three students (3 is a hard maximum).

The following exercise is **mandatory**:

Exercise Project.1:
You are given an $m \times n$-*board*, consisting of m rows and n columns, where each tile on the board is colored either black or white. Each transformation of the board that shifts a row or a column by a number of positions $s \in \mathbb{N}$ in one of the two possible directions such that tiles that are moved off the board are reinserted on the other side is called an *operation*, e. g., shifting the first column three positions down is an operation. Consider the following problem:

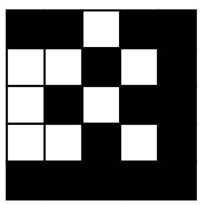
Problem: [MADRAGON]
Input: Two $m \times n$-boards A and B and a number $k \in \mathbb{N}$.
Output: YES if there is a sequence of at most k operations (each of which may shift by an arbitrary number of positions) that transforms A into B and NO otherwise.

What you have to do:

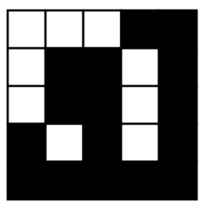
a) Determine whether the answer for the following MADRAGON instance is YES or NO.

   Let $k = 3$ and let the boards be given by

   Start board A:



   Goal board B:

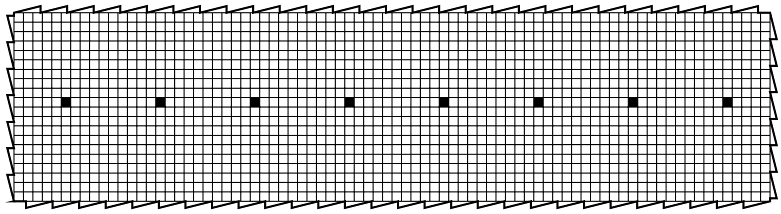


   The goal board is part of the famous Arecibo message.

b) Show that MADRAGON is in $\mathcal{NP}$.

c) Show that MADRAGON is $\mathcal{NP}$-complete. As reference problem you must select a problem from the list of $\mathcal{NP}$-complete problems given below. Note that there may be many different approaches to prove $\mathcal{NP}$-completeness.

Hint: we recommend that part of your starting board of your chosen problem's transformed instance should be similar to this:



To prove the correctness of your transformation, note also the following simple property: if you want to move k tiles from a certain row to other rows by using column operations only, you need at least k such operations (analogously this holds for moving tiles from a column by using row operations only).

d) Find an algorithm which solves the optimizing version of the problem, i. e., given two boards A and B, it determines and outputs a minimum-length sequence of operations that transform board A into board B. The algorithm is allowed have exponential worst-case running time but may use "smart"/"heuristic" techniques to deal faster with some instances.

Describe in words how the algorithm works.

e) Prove the worst-case running time of your algorithm.

f) Some problem instances are given on Campusnet as text files in the following format (with file name extension `.mad`):

The file is an ASCII file consisting of lines separated by the line-feed symbol; besides the line-feed, the only allowed characters in the file are numbers $\{\texttt{0}, \texttt{1}, \ldots, \texttt{9}\}$ and the two lower-case letters `b` and `w`.

The first line of the file contains the number m of rows, the second line the number n of columns and the third line contains the number k of allowed operations, all three in decimal representation. The subsequent m lines encode the starting board A, where each line encodes a row of A. The lines are given in the same order as the rows of A. Each of these lines consists of a string of length n, consisting of either `b` (black) or `w` (white), such that the color-characters follow the same order as the tiles' colors in the corresponding row.

For example, the MADRAGON instance from part a) could be stored as follows:

```
5
5
3
bbwbb
wwbwb
wbwbb
wwbwb
bbbbb
wwwbb
wbbwb
wbbwb
bwbwb
bbbbb
```

Implement the algorithm you developed in Part d) and run it on some MADRAGON instances.

The program including the source code and an instruction how to execute it on an MADRAGRON instance file has to be delivered to the teaching assistant. Accepted programming languages are Java, C, C++, C#. Other languages have to be agreed upon with the teaching assistant.
Your programs will be run on some MADRAGON instances.

The three blocks [a),b),c)], [d),e)], and [f)] have approximately equal weights in the grading.

List of $\mathcal{NP}$-complete problems to choose from.

Problem: [MINIMUMDEGREESPANNINGTREE]
Input: An undirected graph $G = (V, E)$ and a natural number k.
Output: YES if there is a spanning tree T in which every node has degree at most k. NO otherwise.

Problem: [MAXIMUMCOMMONINDUCEDSUBGRAPH]
Input: Two undirected graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ and a natural number k.
Output: YES if there is a common induced subgraph of cardinality k, i. e., subsets $V_1{}' \subseteq V_1$ and $V_2{}' \subseteq V_2$ such that $|V_1'| = |V_2'| = k$, and the subgraph of G_1 induced by V_1' and the subgraph of G_2 induced by V_2' are isomorphic. NO otherwise.

Problem: [TRIPLEBINPACKING]
Input: $n = 3k$ items with sizes $s_1, \dots, s_n \in \mathbb{N}$. There are k bins of size $(s_1 + \cdots + s_n)/k \in \mathbb{N}$ each.
Output: YES if there is a way of putting exactly 3 items into every bin without overloading any of the bins. NO otherwise.

Problem: [MINIMUMCLIQUECOVER]
Input: An undirected graph $G = (V, E)$ and a natural number k.
Output: YES if there is clique cover for G of size at most k. That is, a collection $V_1, V_2, \dots, V_k$ of not necessarily disjoint subsets of V, such that each V_i induces a complete subgraph of G and such that for each edge $\{u, v\} \in E$ there is some V_i that contains both u and v. NO otherwise.

Problem: [LONGESTCOMMONSUBSEQUENCE]
Input: A sequence $\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_n$ of strings over an alphabet Σ and a natural number B.
Output: YES if there is a string $\boldsymbol{x}$ over Σ of length B which is a subsequence of all $\boldsymbol{w}_i$. NO otherwise.

Formally, we say that $\boldsymbol{x} = x_1 x_2 \cdots x_{\ell_x}$ is a *subsequence* of $\boldsymbol{w} = w_1 w_2 \cdots w_{\ell_w}$ if there is a strictly increasing sequence of indices i_j, $1 \le j \le \ell_x$, such that for all $j = 1, 2, \dots, \ell_x$ we have $x_j = w_{i_j}$.

Problem: [MINIMUMRECTANGLETILING]
Input: An $n \times n$ array A of non-negative integers, natural numbers k and B.
Output: YES if there is a partition of A into k non-overlapping rectangular sub-arrays such that the sum of the entries every sub-array is at most B. NO otherwise.

Problem: [PARTITIONBYPAIRS]
Input: A sequence $S = (s_1, s_2, \dots, s_{2n})$ of natural numbers.
Output: YES if there is a subset $A \subseteq \{1, \dots, 2n\}$ choosing exactly one from each pair $(2i-1, 2i)$, where $i \in \{1, \dots, n\}$, such that $\sum_{i \in A} s_i = \sum_{i \in \{1,\dots,2n\} \setminus A} s_i$. NO otherwise.

Problem: [1In3Satisfiability]
Input: A set of clauses $C = \{c_1, \dots, c_k\}$ over n boolean variables $x_1, \dots, x_n$, where every clause contains exactly three literals.
Output: YES if there is a satisfying assignment such that every clause has exactly one true literal, i. e., if there is an assignment

$$a\colon \{x_1, \dots, x_n\} \to \{0, 1\}$$

such that every clause c_j is satisfied and no clause has two or three true literals. NO otherwise.

Background
The Madragon-problem is adapted from a game of the same name that was developed and released by Martin J. Klein in 1995 and distributed by mailorder in Germany (a screenshot is given in Figure 1). In the original game, the boards were of size 9×9 and players could beat their personal record with respect to the time needed to solve it.

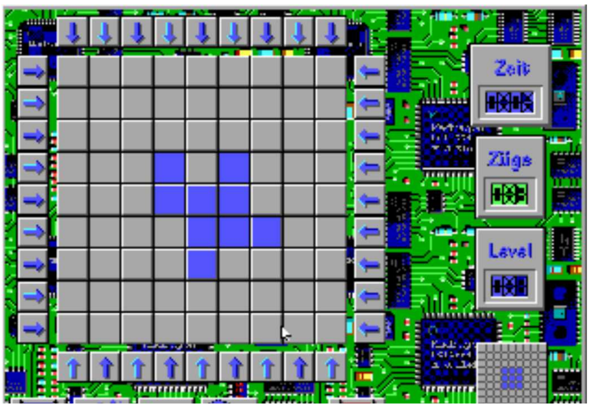


Figure 1: Screenshot of the original game Madragon, developed by Martin J. Klein and released in 1995 for DOS systems.

End of Exercise 1