Problem Set 7
CSCE 411
Andreas Klappenecker

The assignment is due on Wednesday, Nov 09, 2011, before class.

Let n be a positive integer. *Comparative Sudoku* is a puzzle that consists of an $n^2 \times n^2$ grid that is subdivided into
(a) n^2 rows,
(b) n^2 columns, and
(c) n^2 disjoint regions of $n \times n$ squares.
Within each square region, there is a comparison operator $<$ or $>$ between any two adjacent cells. The goal is to place integers from the set $[n^2] := \{1, 2, \ldots, n^2\}$ into the grid such that each row, each column, and each square region contains each element of $[n^2]$ precisely once, and such that all inequalities between adjacent cells within the squares are satisfied.

Example 1. *For $n = 2$, consider the $2^2 \times 2^2$ comparative Sudoku*

.	$<$	.	.	$<$	.
$\wedge$		$\wedge$	$\vee$		$\vee$
.	$<$	.	.	$<$	.
.	$<$	.	.	$>$	.
$\wedge$		$\vee$	$\vee$		$\wedge$
.	$>$	.	.	$<$	.

where the dots represent the cells. This puzzle can be solved as follows:

1	$<$	2	3	$<$	4
$\wedge$		$\wedge$	$\vee$		$\vee$
3	$<$	4	1	$<$	2
2	$<$	3	4	$>$	1
$\wedge$		$\vee$	$\vee$		$\wedge$
4	$>$	1	2	$<$	3

Exercise 1 (30 points)**.** *Describe the design of a backtracking algorithm that is able to find the solution of a comparative Sudoku puzzle if one exists; otherwise, the algorithm should return "does not have a solution".*

The solution to the puzzle should be stored in an array

```
p[0..n**2-1][0..n**2-1]
```

Explain how the comparison operators of the rows, and of the columns are stored in your algorithm.

The algorithm should proceed filling creating the potential solutions in the order

```
p[0][0], p[0][1], ..., p[0][n**2-1], p[1][0], ...
```

that is, the solution should be successively obtained row-by-row, filling each row from the left. When a putative value for the next entry is created, the contraints should be checked in the following order:

(i) row entries must be distinct
(ii) column entries must be distinct
(iii) square region must contain distinct elements
(iv) specified comparisons within each square must be obeyed

Keep your description of your algorithm sufficiently abstract, similar to the style used in the book.

[Hint: If you are unfamiliar with backtracking, then you should study the backtracking solution to the N queens problem before attempting this exercise.]

Exercise 2 (20 points)**.** *The file format specifies n, then the comparison operators of the rows, and then the comparison operators of the columns. For example:*

```
n=2
rows:
< <
< <
< >
> <
columns:
< <
< >
> >
> <
```

This specifies the puzzle given in the previous example. The $n(n-1)$ comparison operators of each row are given in n blocks of $n-1$ comparison operators ($<$ or $>$), and the blocks are separated by spaces. Rows can be separated linefeed, carriage return, or a combination of both.

Implement a procedure to read in such a file. Furthermore, implement a procedure to print a solution (or partial solution) in readable form. Follow the example:

```
---------
|1<2|3<4|
|^ ^|v v|
|3<4|1<2|
---------
|2<3|4>1|
|^ v|v ^|
|4>1|2<3|
---------
```

You will need these two procedures for this as well as future homeworks.

Exercise 3 (30 points)**.** *Implement your algorithm in C, C++, Java, or Ruby. Make sure that your program is well-structured and well-documented. Attach a printout to your homework submission.*

Exercise 4 (20 points)**.** *Give your solution to the challenge problems that will be posted shortly before the deadline.*

The same problem will be used in our programming contest. However, you will have to change the algorithm, since the approach taken here is too slow.