Assignment 2

Due Date: Oct 19th 2012

This assignment has 2 problems. The 1st one on alpha-beta search is a written one and the 2nd one on CSP is a programming problem.

I. (Written) Apply the alpha-beta algorithm on the minimax tree shown in Figure 1.

- Mark the roots of subtrees that will be pruned.
- Also at every non-pruned node show the alpha-beta value when reaching that node for the first time and the updated alpha-beta value every time the algorithm backs up to that node and the final value of the node upon exiting from the node to its parent.

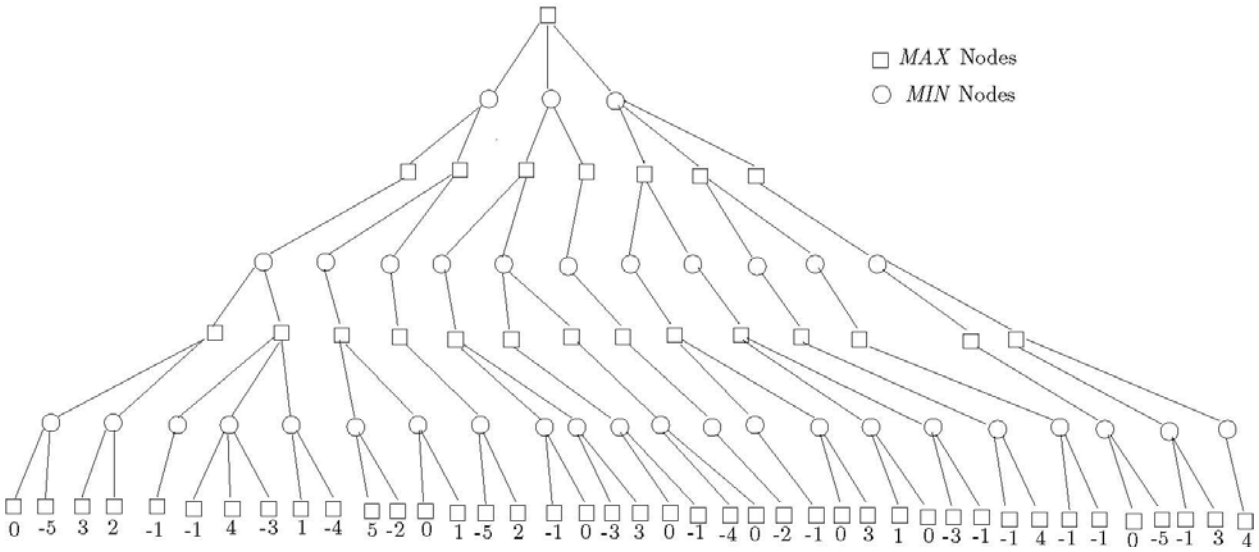

Figure 1

II. (Programming) Sudoku is a number placement puzzle. In this puzzle you are given an N x N grid of cells. The grid itself is composed of M x K sub-grids. You can place a single digit, drawn from 1 to N, in any cell. Initially the grid will have some of its cells partially filled. The objective of the puzzle is to complete the grid so that:

1. Every cell contains a digit.
2. No digit appears twice in any row, column of the N x N grid or in any row, column of any of the M x K sub-grid.

Figure 2 (a) below is a 12 x 12 Sudoku puzzle made up of 3 x 4 sub-grids and Figure 2 (b) is the solution to this puzzle.

	11								4		
7			2	6			3	5		11	
	6	9		1	12			7		10	
	4	1					10	8		6	
	8		9				12	10			
2				11		1			9		
		8			2		4				7
			3	5				12		4	
	7		4	12					6	8	
	5		8			10	7		11	1	
	1		11	3			5	2			6
		3								12	

Figure 2(a)

8	11	12	1	7	10	5	2	6	4	3	9
7	10	4	2	6	8	9	3	5	12	11	1
3	6	9	5	1	12	4	11	7	2	10	8
11	4	1	12	9	5	2	10	8	7	6	3
6	8	5	9	4	3	7	12	10	1	2	11
2	3	7	10	11	6	1	8	4	9	5	12
5	12	8	6	10	2	11	4	1	3	9	7
1	9	11	3	5	7	8	6	12	10	4	2
10	7	2	4	12	1	3	9	11	6	8	5
12	5	6	8	2	9	10	7	3	11	1	4
9	1	10	11	3	4	12	5	2	8	7	6
4	2	3	7	8	11	6	1	9	5	12	10

Figure 2(b)

Your assignment is to write a program to solve the general Sudoku puzzle using finite domain CSP methods discussed in the class.

The input to your program will be a file formatted as follows:

1. The first line will have three integers that fix the values for N, M and K in that order.
2. The next N lines describe the board configuration – one per row. An empty cell will be denoted by _.

So N, M, K and the first two rows in Figure 2(a) will be represented like this:

12, 3, 4
_,11,_,_,_,_,_,_,4,_,_
7,_,_,2,6,_,_,3,5,_,11,_

You should implement:

1. Backtracking
2. Backtracking + MRV heuristic
3. Backtracking + MRV + Forward Checking
4. Backtracking + MRV + Constraint Propagation

Given an input instance you will generate an output for each of the 4 implementations. Each output will be a solved puzzle. The first N lines generated by the output describe the solved puzzle with each line describing a row of the completed puzzle. The last line should be the number of consistency checks done to arrive at the solution.

We will provide you with some test data. Based on these test files analyze the pros and cons of these 4 implementations.

Notes:

1. Your program should be written in Java.
2. You can work in pairs of two

3. There are numerous sources on the Web for the Sudoku puzzle. You are encouraged to consult them to gain a good understanding of the puzzle.

Submission:
On or before due date you should email to the TA a zip file containing:

- Source code with good documentation
- A report with a critical analysis of the 4 implementation methods
- Answer to the written problem

You have until midnight of the due date to email the zip file. You should sign up for a demo with the TA. On the demo date you will be given your source files that you should compile and demo to the TA. Your program will be tested on different instances of the puzzle.