CENG317 – Principles of Artificial Intelligence

Assignment #3

Due: 30.11.2023 @13:29

ATTENTION:

- Programs which contain **errors** will **NOT** be graded.
- In all of the programs, use **COMMENTS** where necessary.
- Code blocks inside comments will **NOT** be graded.
- Send only .java files and in a .zip file named as **"hw3_yourName_secondIndividualName.zip"**.

Disciplinary action to be taken against cheating is arranged by the Rules and Regulations Governing Student Disciplinary Actions in Institutions of Higher Education.

QUESTIONS (100pts)

1. (100pts) Solve the following 4-star sudoku by using Simulated Annealing:

		6						
	8			5	4	2		
	4			9			7	
		7	9			3		
				8		4		
6						1		
2		3						1
			5				4	
		8	3			5		2

Steps:

- first guessing a solution at random

```
Initial Board:
-------------------------
| 7 2 6 | 3 7 1 | 9 1 5 |
| 9 8 1 | 8 5 4 | 2 8 4 |
| 3 4 5 | 2 9 6 | 3 7 6 |
-------------------------
| 2 1 7 | 9 6 7 | 3 9 8 |
| 5 8 9 | 4 8 1 | 4 7 6 |
| 6 4 3 | 2 5 3 | 1 5 2 |
-------------------------
| 2 6 3 | 2 6 4 | 7 6 1 |
| 9 7 5 | 5 1 8 | 3 4 8 |
| 4 1 8 | 3 7 9 | 5 9 2 |
-------------------------
Initial Fault Score: 42
```

- filling in the empty cells above with random digits between 1 and 9. (for the sake of the simplicity, you may fill each 3x3 squares correctly)
- scoring this solution by counting the number of digits duplicated in all the rows, columns and blocks.
- evaluating a number of candidate new solutions by tweaking one of the free digits, and score those.
- then selecting one of the candidate solutions at random for the next step, weighted by the change in the score.

SA-Pseudo Code:

```
Create random initial solution γ
E_old=cost(γ);
for(temp=temp_max; temp>=temp_min; temp=next_temp(temp))
{
	for(i=0; i<i_max; i++ )
	{
		succesor_func(γ); //this is a randomized function
		E_new=cost(γ);
		delta=E_new-E_old;
		if(delta>0)
			if(random() >= exp(-delta/K*temp);
				undo_func(γ); //rejected bad move
			else
				E_old=E_new  //accepted bad move
		else
			E_old=E_new; //always accept good moves
	}
}
```

You can complete this task in groups of 2 people. Since you will be demonstrating the result in the class, it is recommended that you do this together. Try to be informative in console outputs. Make an executable .jar file and demonstrate your results and codes in class.

There are many examples on the web, you can get inspiration from them. However, copying/pasting is not allowed. You are expected to fully disclose all the work you have done.

Example Run:

```
Initial Board:
-------------------------
| 7 2 6 | 3 7 1 | 9 1 5 |
| 9 8 1 | 8 5 4 | 2 8 4 |
| 3 4 5 | 2 9 6 | 3 7 6 |
-------------------------
| 2 1 7 | 9 6 7 | 3 9 8 |
| 5 8 9 | 4 8 1 | 4 7 6 |
| 6 4 3 | 2 5 3 | 1 5 2 |
-------------------------
| 2 6 3 | 2 6 4 | 7 6 1 |
| 9 7 5 | 5 1 8 | 3 4 8 |
| 4 1 8 | 3 7 9 | 5 9 2 |
-------------------------
Initial Fault Score: 42

Iteration: 100 - Fault Score: 27
Iteration: 200 - Fault Score: 21
Iteration: 300 - Fault Score: 14
Iteration: 400 - Fault Score: 12
Iteration: 500 - Fault Score: 10
Iteration: 600 - Fault Score: 8
```

```
Iteration: 700 - Fault Score: 8
Iteration: 800 - Fault Score: 11
Iteration: 25000 - Fault Score: 2
Iteration: 50000 - Fault Score: 4
Iteration: 75000 - Fault Score: 6
Iteration: 100000 - Fault Score: 2
Iteration: 125000 - Fault Score: 5
Iteration: 150000 - Fault Score: 4
Iteration: 175000 - Fault Score: 4
Iteration: 200000 - Fault Score: 4

Fault Score of final Solution: 0
Found in 278788. iteration

Final Solution:
-------------------------
| 5 9 6 | 2 7 3 | 8 1 4 |
| 7 8 1 | 6 5 4 | 2 3 9 |
| 3 4 2 | 8 9 1 | 6 7 5 |
-------------------------
| 8 1 7 | 9 4 2 | 3 5 6 |
| 9 3 5 | 1 8 6 | 4 2 7 |
| 6 2 4 | 7 3 5 | 1 9 8 |
-------------------------
| 2 5 3 | 4 6 7 | 9 8 1 |
| 1 6 9 | 5 2 8 | 7 4 3 |
| 4 7 8 | 3 1 9 | 5 6 2 |
-------------------------
```