CS4200 -- Project 2: N-Queen (n = 8)

Please use Java, Python, or C++

Project Description

Local search algorithms are very efficient in solving n-Queen problems. You are asked to implement the following two algorithms to solve the n-queen problem:

1) Straight-forward steepest-ascent hill climbing as described on page 9 of our lecture "local search".

2) Your choice of one of the following algorithms: simulated annealing search algorithm (page 18 of our "local search" lecture); the genetic algorithm (page 26 of our "local search" lecture); or the MIN-CONFLICTS algorithm (page 42 of our "CSP" lecture).

For analysis, you should generate a large number of n-queen instances (>100) and solve them. Document the percentage of solved problems, search costs and the average running time. Explain why you get such results, for example, why the steepest-ascent hill climbing can only solve about 14% of the problems, or what kind of improvements have you made to make your algorithms more efficient.

If you choose to implement the genetic algorithm, then your goal is to be able to solve the problem. You need to show that your program is able to solve at least three cases. You will also get +10 bonus points for solving the problem using genetic algorithm.

What to Submit?

- Project report (your approach + analysis + findings, <3 pages, word/pdf format).
- Source code + README (how to compile and run your code).
- Program output: sample solutions (at least 3 different solutions). Just need to show the final configuration. Solution path (like the one in project 1) doesn't need to be included.
- Please create a folder called "yourname_4200p2" that includes all the required files and generate a zip file called "yourname_4200p2.zip".
- Please submit your zip file through canvas.