CS4200 -- Project 3: N-Queen (n = 25) using Min Conflicts Algorithm
You are only allowed to use Java, or C++

Project Description:

Local search algorithms are very efficient in solving n-Queen problems (where n = 25). You must implement the following two algorithms to solve the n-queen problem:

1) Min Conflicts Algorithm

You should implement the CSP problem as a CSP problem, not as a standard search problem. .

Analysis:

For analysis, you should generate a large number of n-queens instances (>400) and solve them. Record the success ratio and average time for a solution. Compare the speed and success rate of you Min Conflicts implementation to the performance of your programs from project 2.

User Interface:

The program will start and immediately begin solving the problem. The output will be the name of the algorithm, a solution, the solution's fitness, and the runtime for each algorithm.

What to Submit:

Project report (your approach + analysis + findings, <3 page in pdf format).

Source code + README (how to compile and run your code. **Do not assume that the grader will use your IDE. You need to instruct the grader on how to compile and run your program from the command line**).

Program output: sample solutions (at least 3 different solutions). You just need to show the final configuration, the solution path doesn't need to be included; remember, local search.

How to Submit:

Create a folder called, "lastname_firstname_4200p3", that includes all of the required files, from which, you should generate a zip file called "lastname_firstname_4200p3.zip". For example, if Jane Doe was submitting a project, she would name the folder doe_jane_4200p3. The resulting zip file would be named, doe_jane_4200p3.zip. Submit this file via Blackboard before the due date.

No late submissions will be accepted