

CS 2003 Fundamentals of Algorithms and Computer Applications

Project 2: Recursive backtracking, with forward checking, solution to N-Queens

You will modify the `NQueens.java` file (copy it from the `~class_sandip/2003/` directory on `linux.ens.utulsa.edu`) to implement a lookahead-based solution to the N-Queens problem. You also need to copy and compile the `NQueensGUI.java` file from the same directory.

You are required to do the following:

1. Implement the methods in the file that are not implemented and any other helper methods you might use. To test your code, you should compile and run the `NQueensGUI` program (it runs like a normal java application: enter `java NQueensGUI` from a terminal, or Run As a Java application in Eclipse).
2. Add code to write out the number of backtracks (number of times a queen is removed from the board), the number of calls to *isAttacked* and *placeQueens*.
3. Run the program for different input board sizes and plot the number of backtracks and calls to *isAttacked* and *placeQueens* for input values between 4 and 25. You can use any tool to plot your graphs, (OpenOffice, gnuplot are available on the linux.ens server, MS Excel is available on the Windows PCs).

Submission: Submit, via Harvey, the following before the project deadline:

- The complete source code: the file `NQueens.java`, the code used (if any) to generate the graphs, and the data file used to generate the graph.
- A 1–2 page (A4 size, 12 pt, 1" margin) project report containing a short description of your data generation process to plot the above mentioned graphs, and observations or analysis of the plots. In particular, discuss why you see the data trends and also what you conclude about the effectiveness of backtracking in solving the N-Queens problems for large values of N.

Grading Criteria: The approximate grade composition for this project is as follows:

<code>setQueenAndMarks</code> method	10 points
<code>removeQueensAndMarks</code> method	10 points
<code>markForward</code> method (must be tail-recursive)	60 points
Project report	15 points
Comments/style ¹	5 points

Total	100%
-------	------

NOTE: Algorithm or code inefficiency will be penalized!

This assignment must be your own work and you should include your name and student ID in the submissions.