General description of the project

*Linear algorithm for solution n-Queens Completion Problem*

Grigoryan E.

The purpose of this project is the Research and development of an effective algorithm for solving ***n-Queens Completion Problem***. The study (with various intensities and interruptions) was conducted during 2018-2019 years and took about a year and a half. This study did not have financial support. The results of the study are published in arxiv.org in the article: *Grigoryan E., Linear algorithm for solution n-Queens Completion problem,* [*https://arxiv.org/abs/1912.05935*](https://arxiv.org/abs/1912.05935). The Russian version of the article is published on the ***habr.com*** programmer community website: <https://habr.com/ru/post/483036/>

The task has a simple mathematical formulation. There is a composition of ***k*** queens that are consistently distributed on a ***n × n*** chessboard. It is required to prove that this composition can be completed to a full solution, and give at least one solution, or to prove that such a solution does not exist. A composition of ***k*** queens is considered consistent if the following three conditions of the problem are fulfilled: in each row, each column, and also on the left and right diagonals passing through the cells where the queens are located, no more than one queen is located. In the algorithmic formulation of this problem, instead of the phrase “it is required to prove that ...” it will be written “it is required to develop an algorithm that ...”

In 2017, in a large article: *C. Gent, I.-P. Jefferson and P. Nightingale. Complexity of n-queens completion. Journal of Artificial Intelligence Research, 59, 815–848, 2017*, <https://jair.org/index.php/jair/article/view/11079/26262>, it has been proved that this problem belongs to the ***NP-Complete*** set. It is assumed that the solution to this problem may open the way for solving some other problems from the NP-Complete set.

**Checking the correctness of the program**

Any program (algorithm) has a simple property - it either works or not. To test this, just run it for execution. To check this algorithm, it is necessary to select the size of the chessboard (***n***), form a random composition of size ***k (1 <k <n)***, and start the completion program. As a result, the program must complete the composition to a full solution, or decide that the composition cannot be completed. However, one arrangement of queens is not enough. In order to check whether the time complexity of completion program is linear, it is necessary:

- to form a large sample of random compositions for various values of ***n***,

- run the completion program for all compositions of each sample,

- determine for each sample the average completion time.

If we divide the average time by ***n***, we get the reduced time, i.e. the average time required to correctly position the queen in one position. Having plotted the dependence of the reduced time on the size of the chessboard, we can establish the time complexity of this algorithm. Obviously, the time complexity of the algorithm will be linear if the graph is presented as a straight line parallel to the abscissa axis.

To test the algorithm, you do not have to write any code. In addition to the main program:

*Solution\_n\_Queens\_Completion\_Problem.m* – completion an arbitrary composition to full solution, or making a decision that this composition has no decision,

two more programs have been prepared:

*Generarion\_k\_Queens\_Composition.m* – generating a random composition of size ***k*** for an arbitrary chessboard of size ***n x n***,

*Validation\_n\_Queens\_Solution.m* – checking the correctness of the solution of the ***n-Queens Problem,*** or the correctness of the composition of ***k*** queens.

They work very fast. For example, for a chessboard with the size ***1000 x 1000*** cells, the total time, which on average, is necessary for:

- generation of an arbitrary composition (***0.0015 s.***),

- complete this formed composition (***0.0622 s***.),

- and verification of the correctness of the solution (***0.0003 s***.),

does not exceed ***0.1*** seconds (excluding the time it takes to save the results or download data).

The source code of the programs is presented with very detailed comments and is designed for a wide range of users. If you will read the article in ***arxiv.org*** and the comments in the source code of the program, and this did not help to understand any part of the code, or the logic of the executable procedure, then write me ([ericgrig@gmail.com](mailto:ericgrig@gmail.com)), I will certainly find the opportunity to answer. For some questions that may arise, I have prepared answers, and they are presented in the *Q & A\_**n-Queens Completion Problem.docx* . file. All new questions and answers to them will also be posted in this file.

The algorithm does not depend on what language it is written in. The choice of the scripting language Matlab is mainly related to the convenience of modeling. If you synchronously translate this program into ***C*** or ***C ++***, then it will work faster. I will be glad if someone takes the initiative and translates this program into other languages. For my part, I can find an opportunity to provide advisory support.

The first version of the program was written pedagogically correctly, with the allocation of functions, where there are repeating, logically connected, sections of code. Then the algorithm was improved, and some parts of the code changed, and accordingly those parts of the code that were associated with it changed. At the initial stage, this was repeated quite often and, over time, these functions “disappeared” in the program. The algorithm that is presented here is the fifth version of the program and there are no functions in it. However, in the text with an “unarmed look” sections of repeating codes are visible, which, I hope, will be correctly taken into account when translating a program into another programming language.

For the convenience of readers, I have prepared all the related information in three languages: English, French, Russian. I would be deeply grateful to anyone who can correctly translate related information and comments in the source code into other languages.

Grigoryan Eros

 October, 2020, Marseille, France, [ericgrig@gmail.com](mailto:ericgrig@gmail.com)