Общее описание проекта

*Linear algorithm for solution n-Queens Completion Problem*

Grigoryan E.

Цель данного проекта – Исследование и разработка эффективного алгоритма решения ***n-Queens Completion Problem.*** Исследование (с различной интенсивностью и перерывами) проводилось в течение 2018-2019 гг. и заняло примерно полтора года. Данное исследование не имело финансовой поддержки. Результаты исследования опубликованы в *arxiv.org* в статье: *Grigoryan E., Linear algorithm for solution n-Queens Completion problem,* [*https://arxiv.org/abs/1912.05935*](https://arxiv.org/abs/1912.05935) *.* Русскоязычная версия статьи опубликована на сайте сообщества программистов habr.com: <https://habr.com/ru/post/483036/>

Задача имеет простую математическую формулировку. Имеется композиция из ***k*** ферзей, которые непротиворечиво распределены на шахматной доске размером ***n × n***. Требуется доказать, что данная композиция может быть комплектована до полного решения, и привести хотя бы одно решение, или доказать, что такого решения не существует. Композиция из ***k*** ферзей считается непротиворечивой, если выполняются следующие три условия задачи: в каждой строке, каждом столбце, а также на левой и правой диагоналях, проходящих через ячейки, где расположены ферзи, не расположено более одного ферзя. В алгоритмической формулировке данной задачи, вместо фразы «требуется доказать, что …» следует записать «требуется разработать алгоритм, который…»

В 2017 году, в большой статье: *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>, было доказано, что данная задача относится к множеству ***NP-Complete***. Предполагается, что решение данной проблемы, возможно, откроет путь для решения некоторых других задач из множества ***NP-Complete***.

**Проверка правильности работы программы**

Любая программа (алгоритм) имеет простое свойство – она или работает, или нет. Чтобы проверить это, достаточно просто запустить на исполнение. В данном проекте, для этого нужно выбрать размер шахматной доски (***n***), сформировать случайную композицию размера ***k (1< k < n)***, и запустить программу комплектации. В результате, программа должна комплектовать композицию до полного решения, либо принять решение, что композиция не может быть комплектована. Однако, одной композиции мало. Чтобы проверить, является ли временная сложность алгоритма линейной, необходимо:

- формировать большую выборку случайных композиций для различных значений ***n***,

- запустить программу для комплектации всех композиций каждой выборки,

- определить среднее время комплектации композиций каждой выборки.

Если поделить ***среднее время*** на **n**, то получим ***приведенное время***, т.е. среднее время, необходимое для правильного расположения ферзя на одну позицию. Построив график зависимости приведенного времени от размера шахматной доски, можно установить временную сложность данного алгоритма. Очевидно, что ***временная сложность алгоритма будет линейной, если график будет представлен в виде прямой, параллельной оси абсцисс***. Чтобы протестировать алгоритм, не придется писать какой либо код, все уже подготовлено. Кроме основной программы:  
  
*Solution\_n\_Queens\_Completion\_Problem.m* – комплектация произвольной композиции до полного решения, либо принятие решение, что данная композиция не имеет решения,

подготовлены еще две программы:

*Generarion\_k\_Queens\_Composition.m* – генерация случайной композиции размера ***k*** для произвольной шахматной доски размера ***n x n*** *,*  
  
  
*Validation\_n\_Queens\_Problem\_Solution.m* – проверка правильности решения ***n-Queens Problem***, либо правильности композиции из ***k*** ферзей.  
  
Они работают очень быстро. Например, для шахматной доски, размер которой составляет ***1000 х 1000*** клеток, общее время, которое в среднем, необходимо для:

- генерации произвольной композиции (***0.0015 с***.),

- комплектации сформированной композиции (***0.0622 с***.),

- и проверки правильности полученного решения (***0.0003 с***.),

не превышает **0.1** секунды (исключая время, которое необходимо для сохранения полученных результатов, или загрузки данных).

Исходный код программ представлен с очень подробными комментариями и рассчитан на широкий круг пользователей. Если, вы прочли статью в ***arxiv.org*** и комментарии в исходном коде программы, и это не помогло понять какой-то участок кода, или логику исполняемой процедуры, то напишите мне ([ericgrig@gmail.com](mailto:ericgrig@gmail.com)), я обязательно найду возможность ответить. Для некоторых вопросов, которые могут возникнуть, я подготовил ответы, и они представлены в файле *Q & A\_**n-Queens Completion Problem.docx* . Все новые вопросы и ответы к ним, также будут размещены в этом файле.

Алгоритм не зависит от того, на каком языке он написан. Выбор скриптового языка Matlab, в основном связан с удобствами моделирования. Если синхронно перевести данную программу на ***C*** или ***C++***, то она будет работать быстрее. Буду рад, если найдутся талантливые люди, желающие перевести данную программу на другие языки. Со своей стороны, я смогу найти возможность оказать консультативную поддержку.

Первая версия программы была написана педагогически правильно, с выделением функций, там, где есть повторяющиеся, логически связанные, участки кода. Потом алгоритм усовершенствовался, и некоторые участки кода менялись, и, соответственно изменялись те участки кода, которые с этим связаны. На начальном этапе это повторялось довольно часто и, со временем эти функции «растворились» в программе. Алгоритм, который представлен здесь – это пятая версия программы и функций в ней нет. Однако в тексте «не вооруженным взглядом» видны участки повторяющихся кодов, которые, я надеюсь, будут правильно учтены, при переводе программы на другой язык программирования.

Для удобства читателей, я подготовил всю сопутствующую информацию на трех языках: английском, французском, русском. Буду глубоко признателен всем, кто сможет корректно перевести сопутствующую информацию и комментарии в исходном коде на другие языки.

Grigoryan Eros

октябрь, 2020, Марсель, Франция, [ericgrig@gmail.com](mailto:ericgrig@gmail.com)