

АЛГОРИТМ ДЛЯ СБОРКИ КУБИКА РУБИКА 2×2 НА $C\#$ *

Е.П. МИКОВ¹, В.А. БОНДАРЬ², А.А. АЛИМОВ³

¹ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматизи. E-mail: bondar.vale@mail.ru

² 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматизи. E-mail: mikov.e.p@gmail.com

³ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматизи. E-mail: alimov_sanya@mail.ru

Известно, что существует целая совокупность математических методов для изучения свойств кубика Рубика с абстрактно-математической точки зрения, а дисциплина, изучающая эти закономерности, именуется как математика кубика Рубика. Эта математика изучает различные алгоритмы сборки кубика и оценивает их. В основу своего научного подхода она задеиствует теорию графов, теорию групп, а также теории вычислимости и комбинаторики. Начиная с середины XX века многие математики, программисты и любители стремились найти идеальный алгоритм, чтобы на практике за минимальное число ходов собирать кубик Рубика. Существует множество алгоритмов, предназначенных для перевода кубика Рубика из произвольной конфигурации в его конечную конфигурацию, однако следует упомянуть, что в 2010 году программистами из компании Google было выдвинуто строгое математическое обоснование того, что для перевода кубика Рубика из произвольной конфигурации в его собранный вид достаточно не более чем 20 поворотов граней. Для того чтобы выполнить необходимый объем вычислений, потребовалось 35 лет процессорного времени, а продолжительность вычислений составила несколько недель.

Справедливо утверждение, что наиболее приоритетной задачей при разработке алгоритма кубика Рубика является создание быстродействующего и оптимизированного метода сборки. Для решения этой задачи существует много алгоритмов, но программа, разработанная нами, является уникальным вариантом реализации метода сборки кубика Рубика 2×2 на языке программирования $C\#$.

Ключевые слова: $C\#$, программа, алгоритм Бога, кубик Рубика, 2×2 , алгоритмы, .NET, нейронная сеть

* Статья получена 30 августа 2018 г.

ВВЕДЕНИЕ

В ходе обсуждения возможных способов решения кубика Рубика возник интересный алгоритм реализации этой задачи, получивший название «алгоритм Бога». Под алгоритмом Бога головоломки подразумевается любой алгоритм, который позволяет получить решение головоломки, содержащее минимально возможное число ходов (оптимальное решение), начиная с любой заданной конфигурации, соответственно данный термин может также быть использован в отношении других перестановочных головоломок.

1. ЦЕЛЬ ПРОДЕЛАННОЙ РАБОТЫ И АЛГОРИТМ БОГА

Целью нашей совместной работы являлось создание алгоритма для нахождения минимально возможного числа ходов сборки кубика Рубика, начиная с любой случайно заданной конфигурации.

Своему появлению термин обязан Джону Конвею, он определил кратчайший путь из любого данного состояния назад к начальному состоянию как алгоритм Бога.

Алгоритм Бога может существовать для головоломок с конечным числом возможных конфигураций и с конечным набором «ходов», допустимых в каждой конфигурации и переводящих текущую конфигурацию в другую. Термин «решить головоломку» означает указать последовательность ходов, переводящих некоторую начальную конфигурацию в некоторую конечную конфигурацию. Оптимально решить головоломку – указать самую короткую последовательность ходов для решения головоломки. Оптимальных решений может быть несколько.

К известным головоломкам, попадающим под это определение, относится и кубик Рубика. Во многих подобных головоломках конечная конфигурация негласно предполагается, в случае кубика Рубика – это одноцветность граней. В этих случаях «собрать головоломку» означает, что требуется для произвольной начальной конфигурации указать последовательность ходов, приводящих к фиксированной конечной конфигурации.

Тогда алгоритм Бога (для данной головоломки) – это алгоритм, который решает головоломку и находит для конфигураций хотя бы одно оптимальное решение.

Некоторые авторы считают, что алгоритм Бога должен также быть практичным, то есть использовать разумный объем памяти и завершаться в разумное время.

Число Бога кубика 2Ч2Ч2 равно 11 ходам, если поворот грани на 180° считается за один ход, или 14 ходам, если поворот грани на 180° считается за два хода. Возможное количество конфигураций кубика 2Ч2Ч2 – **3 674 160**.

У кубика есть 6 сторон, однако очевидно, что, повернув одну сторону, мы сделаем движение, обратное возвратному повороту противоположной стороны, поэтому будем поворачивать только три стороны. Каждую сторону мы назовем буквой: фронтальная F, верхняя U и правая R (рис. 1).

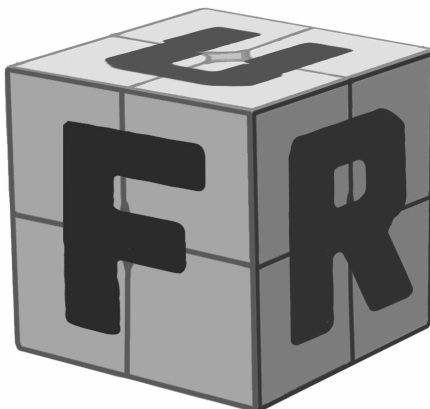


Рис. 1. Кубик с назначенными сторонами

Каждую сторону мы можем поворачивать тремя способами: по часовой стрелке (F), против часовой стрелки (F') и двойной (F²) повороты. Итого мы получаем возможных 9 комбинаций:

$$F F' F^2 U U' U^2 R R' R^2$$

Разных наборов из этих комбинаций получается $9^{11} = 31\,381\,059\,609$. Это довольно большая цифра для жесткого алгоритма, на нахождение всех комбинаций уйдет очень много времени. Поэтому мы используем нейронную сеть – дерево.

Для алгоритма нужен какой-то показатель, по которому будет вычисляться текущее состояние кубика Рубика (коэффициент собранности, в дальнейшем K). За K мы взяли количество правильно повернутых частей (4 части × × 6 сторон = 24), полностью собран – 24, а полностью разобран – 3. Почему 3? Потому что один кубик никогда не меняет своей позиции и всегда находится в правильном положении (три цвета).

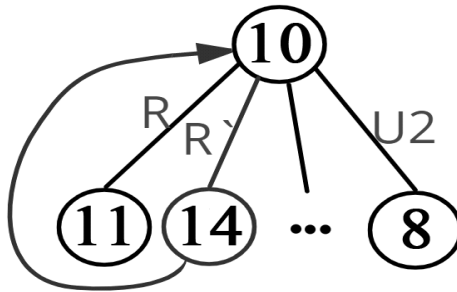


Рис. 2. Дерево

В ходе написания алгоритма мы столкнулись с несколькими проблемами. Во-первых, если мы повернули правую сторону, то в следующей итерации ее крутить нет смысла, так как $R^2 + R = R'$. Ее мы решили, запоминая предыдущие ходы и проверяя впоследствии при выборе пути. Во-вторых, попадание в ямы из восьми итераций. Решение было очевидным – ограничение глубины дерева до 11, именно за столько ходов можно собрать любой кубик Рубика. В-третьих, переполнение памяти из-за рекурсивного подхода. Очищая за собой из буфера всё лишнее, решили проблему переполнения.

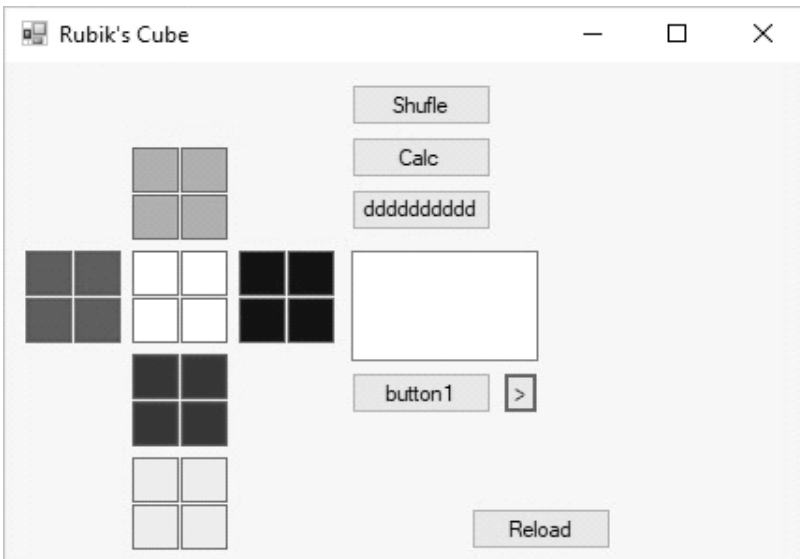


Рис. 3. Наша программа

2. АЛГОРИТМ

1. Получаем K для текущего состояния кубика Рубика.
2. Если $K == 24$, то кубик собран.
3. Если Глубина дерева $== 11$, то выходим на уровень выше и проверяем $K[+1]$.
4. Получаем K кубика после каждого действия и находим максимальный коэффициент собранности ($K[0]$).
5. Идем на шаг 1.

ЗАКЛЮЧЕНИЕ

В данной статье мы постарались наглядно продемонстрировать возможность реализации алгоритма Бога для решения кубика Рубика со сторонами $2 \times 2 \times 2$. Программа, созданная нами в рамках расчетно-графической работы, позволяет найти минимально возможное число ходов, начиная с любой случайно заданной конфигурации. Дальнейшее усложнение этой программы позволит нам реализовать трехмерный интерфейс программы и найти кратчайший маршрут между ними, осуществить возможности работы с кубиками Рубиками со сторонами $3 \times 3 \times 3$ и с большей размерностью. В то же время доподлинно известно, что аналогичная задача для кубика 4Ч4Ч4 на данный момент остается практически неосуществимой. Содержание данной статьи вмещает в себе описание одного из наиболее популярных способов алгоритмизации поиска кратчайшего числа ходов на примере алгоритма Бога. Тема данной работы является актуальной в сфере решения головоломок с конечным числом возможных конфигураций.

БЛАГОДАРНОСТИ

Авторы выражают искреннюю благодарность профессору кафедры автоматки А.А. Воеводе за помощь при написании работы, а также за полезное обсуждение полученных результатов.

СПИСОК ЛИТЕРАТУРЫ

1. Pages and scripts are written by JaapScherphuis, 1999–2015 [Electronic resource]. – URL: <http://www.jaapsch.net/puzzles/cube2.htm> (accessed: 15.03.2019).
2. Pages and scripts are written by JaapScherphuis, 1999–2015 [Electronic resource]. – URL: <http://www.jaapsch.net/puzzles/pyraminx.htm> (accessed: 15.03.2019).

3. Klein E. Résoudre un Rubik's pocket cube avec Neo4j [Electronic resource] // JeffProd. Coding for fun: website. – URL: <https://fr.jeffprod.com/blog/2017/resoudre-un-rubik-pocket-cube-avec-neo4j/> (accessed: 15.03.2019).
4. Алгоритм Бога [Электронный ресурс] // Википедия. – URL: https://ru.wikipedia.org/wiki/Алгоритм_Бога (дата обращения: 15.03.2019).
5. Мильников М. Всем кубикам кубик // Юный техник. – 1982. – № 7. – С. 64.
6. Константинов И. Векторное сложение кубика // Наука и жизнь. – 1999. – № 5. – С. 74.
7. Карасев А. Как научиться собирать кубик Рубика в объеме // Наука и жизнь. – 1999. – № 5. – С. 75.
8. Залгаллер В., Залгаллер С. Венгерский шарнирный кубик // Квант. – 1980. – № 12. – С. 17–21.
9. Дубровский В. Алгоритм волшебного кубика // Квант. – 1982. – № 7. – С. 22–25.
10. Рубик и его кубик: раскрутка, сказочное везение, возвращение [Электронный ресурс]. – URL: <http://www.membrana.ru/> (дата обращения: 15.03.2019).
11. Константинов И. Составляем каталог вращений кубика // Наука и жизнь. – 1985. – № 3–11. – URL: http://www.seoded.ru/downloads/cube/NKJ_1985-03.pdf (дата обращения: 15.03.2019).
12. Кубик Рубика и проблема Хигмана / 20-я летняя конференция международного математического турнира городов. – URL: <http://olympiads.mcsme.ru/lktg/2008/2/2-1ru.pdf> (дата обращения: 15.03.2019).

Миков Егор Петрович, студент факультета автоматике и вычислительной техники Новосибирского государственного технического университета. E-mail: mikov.e.p@gmail.com

Бондарь Валерий Александрович, студент факультета автоматике и вычислительной техники Новосибирского государственного технического университета. E-mail: bondar.vale@mail.ru

Алимов Александр Александрович, студент факультета автоматике и вычислительной техники Новосибирского государственного технического университета. E-mail: alimov_sanya@mail.ru

DOI: 10.17212/2307-6879-2018-3-4-83-90

Algorithm for building a Rubik's Cube 2 × 2 on C#*

E.P. Mikov¹, V.A. Bondar², A.A. Alimov³

¹ 630073, Russian Federation, Novosibirsk, Karl Max Avenue, 20, Novosibirsk State University, student of the department of automation. E-mail: bondar.vale@mail.ru

² 630073, Russian Federation, Novosibirsk, Karl Max Avenue, 20, Novosibirsk State University, student of the department of automation. E-mail: mikov.ep@gmail.com

³ 630073, Russian Federation, Novosibirsk, Karl Max Avenue, 20, Novosibirsk State University, student of the department of automation. E-mail: alimov_sanya@mail.ru

It is known that there is a whole set of mathematical methods for studying the properties of the Rubik's cube from an abstract-mathematical point of view, and the discipline studying these laws is referred to as the Rubik-cube mathematics. This math studies different cube assembly algorithms and evaluates them. At the heart of her scientific approach, she uses graph theory, group theory, and the theory of computability and combinatorics.

Since the middle of the 20th century, many mathematicians, programmers and amateurs have sought to find the perfect algorithm in order to practice a Rubik's cube in the minimum number of moves. There are many algorithms designed to translate Rubik's cube from an arbitrary configuration to its final configuration, however, it should be mentioned that in 2010, programmers from Google put forward a rigorous mathematical rationale that to translate a Rubik's cube from an arbitrary configuration into its assembled form no more than 20 facet turns. In order to perform the required amount of calculations, it took 35 years of CPU time, and the duration of the calculations was several weeks.

It is true that the highest priority in the development of the Rubik's Cube algorithm is to create a fast and optimized assembly method. There are many algorithms for solving this problem, but the program developed by us is a unique variant of the implementation of the 2x2 Rubik Cube assembly method in the C# programming language.

Keywords: C#, program, Algorithm of God, Rubik's Cube, 2x2, algorithms, .NET, neural network

REFERENCES

1. *Pages and scripts are written by JaapScherphuis, 1999–2015.* Available at: <http://www.jaapsch.net/puzzles/cube2.htm> (accessed 15.03.2019).
2. *Pages and scripts are written by JaapScherphuis, 1999–2015.* Available at: <http://www.jaapsch.net/puzzles/pyraminx.htm> (accessed 15.03.2019).
3. Klein E. Résoudre un Rubik's pocket cube avec Neo4j. *JeffProd. Coding for fun:* website. Available at: <https://fr.jeffprod.com/blog/2017/resoudre-un-rubik-pocket-cube-avec-neo4j/> (accessed 15.03.2019).

* Received 30 August 2018.

4. Algoritm Boga [God's algorithm]. *Vikipediya* [Wikipedia]. Available at: https://ru.wikipedia.org/wiki/Алгоритм_Бога (accessed 15.03.2019).
5. Myl'nikov M. Vsem kubikam kubik [All cubes cube]. *Yunyi tekhnika*, 1982, no. 7, p. 64. (In Russian).
6. Konstantinov I. Vektornoe slozhenie kubika [Vector addition of a cube]. *Nauka i zhizn' – Science and Life*, 1999, no. 5, p. 74. (In Russian).
7. Karasev A. Kak nauchit'sya sobirat' kubik Rubika v ob'eme [How to learn to collect the Rubik's cube in volume]. *Nauka i zhizn' – Science and Life*, 1999, no. 5, p. 75. (In Russian).
8. Zalgaller V., Zalgaller S. Vengerskii sharnirnyi kubik [Hungarian articulated cube]. *Kvant*, 1980, no. 12, pp. 17–21. (In Russian).
9. Dubrovskii V. Algoritm volshebного kubika [The magic cube algorithm]. *Kvant*, 1982, no. 7, pp. 22–25. (In Russian).
10. *Rubik i ego kubik: raskrutka, skazochnoe vezenie, vozvrashchenie* [Rubik and his cube: promotion, fabulous luck, return]. Available at: <http://www.membrana.ru/> (accessed 15.03.2019).
11. Konstantinov I. Sostavlyаем katalog vrashchenii kubika [A catalog of the rotation of the cube]. *Nauka i zhizn' – Science and Life*, 1985, no. 3–11. (In Russian). Available at: http://www.seoded.ru/downloads/cube/NKJ_1985-03.pdf (accessed 15.03.2019).
12. *Kubik Rubika i problema Khigmana* [Rubik's cube and Higman problem]. 20th summer conference International mathematical Tournament of towns, 2008. Available at: <http://olympiads.mccme.ru/lktg/2008/2/2-1ru.pdf> (accessed 15.03.2019).

Для цитирования:

Миков Е.П., Бондарь В.А., Алимов А.А. Алгоритм для сборки кубика Рубика 2×2 на C# // Сборник научных трудов НГТУ. – 2018. – № 3–4 (93). – С 83–90. – DOI: 10.17212/2307-6879-2018-3-4-83-90.

For citation:

Mikov E.P., Bondar V.A., Alimov A.A. Algoritm dlya sborki kubika Rubika 2×2 na C# [Algorithm for building a Rubik's cube 2x2 on C#]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 3–4 (93), pp. 83–90. DOI: 10.17212/2307-6879-2018-3-4-83-90.