



Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2021. Т. 21, вып. 2. С. 267–277  
*Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2021, vol. 21, iss. 2, pp. 267–277

Научная статья

УДК 519.17

<https://doi.org/10.18500/1816-9791-2021-21-2-267-277>

## Построение цветных графов без проверки на изоморфизм

П. В. Разумовский, М. Б. Абросимов✉

Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, Россия, 410012, г. Саратов, ул. Астраханская, д. 83

**Разумовский Петр Владимирович**, аспирант кафедры теоретических основ компьютерной безопасности и криптографии, [shprotby@gmail.com](mailto:shprotby@gmail.com), <https://orcid.org/0000-0003-3648-3166>

**Абросимов Михаил Борисович**, доктор физико-математических наук, заведующий кафедрой теоретических основ компьютерной безопасности и криптографии, [mic@rambler.ru](mailto:mic@rambler.ru), <https://orcid.org/0000-0002-4473-8790>

**Аннотация.** В работе рассматриваются графы, вершины или ребра которых раскрашены в заданное количество цветов — вершинные и реберные раскраски. Изучение раскрасок графов началось с середины XIX в., однако основное внимание уделяется правильным раскраскам, в которых накладывается ограничение, что цвета смежных вершин или ребер должны быть различными. В данной работе рассматриваются раскраски графов без каких-либо ограничений. Исследуется задача генерации всех неизоморфных вершинных и реберных  $k$ -раскрасок заданного графа без непосредственной проверки на изоморфизм. Задача о нахождении неизоморфных реберных  $k$ -раскрасок сводится к задаче построения всех вершинных  $k$ -раскрасок графа. В основе методов генерации графов без непосредственной проверки на изоморфизм лежит метод канонических представителей. Идея метода состоит в том, что предлагается способ кодирования графов и выбирается некоторое правило, по которому из всех изоморфных графов один граф объявляется каноническим. Строятся все коды и из них оставляются только канонические. Часто в качестве канонического выбирается представитель с наибольшим или наименьшим кодом. На практике порождение всех кодов требует больших вычислительных ресурсов, поэтому используются различные методы оптимизации перебора. В работе предлагаются два алгоритма решения задачи генерации вершинных  $k$ -раскрасок без проверки на изоморфизм методом МакКея и методом Рида – Фараджева. Производится сравнение разработанных алгоритмов генерации раскрасок на двух классах графов — цепях и циклах. Вычислительные эксперименты показывают, что для цепей и циклов алгоритм, построенный на основе метода Рида – Фараджева, работает быстрее.

**Ключевые слова:** граф, разметка графа, раскраска графа, цветной граф, генератор

**Благодарности:** Работа выполнена при поддержке Минобрнауки России в рамках выполнения государственного задания (проект № FSRR-2020-0006).

**Для цитирования:** Разумовский П. В., Абросимов М. Б. Построение цветных графов без проверки на изоморфизм // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2021. Т. 21, вып. 2. С. 267–277. <https://doi.org/10.18500/1816-9791-2021-21-2-267-277>

Статья опубликована на условиях лицензии Creative Commons Attribution License (CC-BY 4.0)



Article

<https://doi.org/10.18500/1816-9791-2021-21-2-267-277>

## Generation of colored graphs with isomorphism rejection

P. V. Razumovsky, M. B. Abrosimov✉

Saratov State University, 83 Astrakhanskaya St., Saratov 410012, Russia

**Peter V. Razumovsky**, [shprotby@gmail.com](mailto:shprotby@gmail.com), <https://orcid.org/0000-0003-3648-3166>

**Mikhail B. Abrosimov**, [mic@rambler.ru](mailto:mic@rambler.ru), <https://orcid.org/0000-0002-4473-8790>

**Abstract.** In the article we consider graphs whose vertices or edges are colored in a given number of colors — vertex and edge colorings. The study of colorings of graphs began in the middle of the 19th century, but the main attention is paid to proper colorings, in which the restriction applies that the colors of adjacent vertices or edges must be different. This paper considers colorings of graphs without any restrictions. We study the problem of generating all non-isomorphic vertex and edge  $k$ -colorings of a given graph without direct checking for isomorphism. The problem of generating non-isomorphic edge  $k$ -colorings is reduced to the problem of constructing all vertex  $k$ -colorings of a graph. Methods for generating graphs without direct checking for isomorphism or isomorphism rejection are based on the method of canonical representatives. The idea of the method is that a method for encoding graphs is proposed and a certain rule is chosen according to which one of all isomorphic graphs is declared canonical. All codes are built and only the canonical ones are accepted. Often, the representative with the largest or smallest code is chosen as the canonical one. In practice, generating all codes requires large computational resources; therefore, various methods of enumeration optimization are used. The paper proposes two algorithms for solving the problem of generating vertex  $k$ -colorings with isomorphism rejection by McKay and Reed – Faradzhev methods. A comparison of the proposed algorithms for generating colorings on two classes of graphs — paths and cycles is made. Computational experiments show that the Reed – Faradzhev method is faster for paths and cycles.

**Keywords:** graph, graph labeling, graph coloring, color graph, generator

**Acknowledgements:** This work was supported by the Ministry of Science and Higher Education of the Russian Federation in the framework of the state task (project No. FSRR-2020-0006).

**For citation:** Razumovsky P. V., Abrosimov M. B. Generation of colored graphs with isomorphism rejection. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2021, vol. 21, iss. 2, pp. 267–277 (in Russian). <https://doi.org/10.18500/1816-9791-2021-21-2-267-277>

This is an open access article distributed under the terms of Creative Commons Attribution License (CC-BY 4.0)

### Введение

Большой интерес в теории графов представляет задача разработки генераторов для построения всех неизоморфных графов определенного типа. Известны генераторы неориентированных графов, ориентированных графов, турниров [1, 2], регулярных графов [3], кубических графов [4] и др. [5] Все эти генераторы позволяют строить графы без меток: вершины построенных графов имеют одинаковый тип (метку или цвет). На практике граф может являться моделью системы, в которой элементы имеют разный тип [6]. Например, для анализа клиент-серверной модели вычислений вершины, соответствующие серверам, будут иметь один тип, а вершины, соответствующие клиентам, — другой тип. В этом случае вершинам приписываются



метки, обозначающие тип, или, что является более традиционным в теории графов, вершины раскрашиваются в разные цвета. Соответствующий граф мы будем называть цветным. Могут накладываться дополнительные ограничения на цвета, назначаемые вершинам. Например, при правильной раскраске смежные вершины должны иметь разные цвета. Изучение раскрасок графов началось с середины XIX в., однако основное внимание уделяется правильным раскраскам в минимальное число цветов [7–9].

В данной статье мы будем рассматривать произвольные раскраски заданного графа, а основная задача состоит в построении всех существенно различных раскрасок заданного графа в  $k$  цветов или  $k$ -раскрасок. В более общем случае связи между элементами также могут иметь разный тип, что приводит к графу, в котором метки или цвета приписываются также и ребрам. В данной работе исследуется генерация всех неизоморфных  $k$ -раскрасок для заданного графа. Мы будем рассматривать как вершинные, так и реберные раскраски. Задача о нахождении неизоморфных реберных  $k$ -раскрасок будет сведена к задаче построения всех вершинных  $k$ -раскрасок графа. Будут рассматриваться раскраски неориентированных графов, однако полученные алгоритмы естественным образом можно перенести и на случай ориентированных графов.

### 1. Основные определения

Прежде чем рассматривать методы построения цветных графов, необходимо дать основные определения, используемые в данной работе. Остальные определения из теории графов можно найти в [10, 11].

Пусть  $G(V, \alpha)$  — граф,  $k \in \mathbb{N}$ . Функция вида  $f : V \rightarrow \{1, \dots, k\}$  называется *вершинной  $k$ -раскраской*  $G$ ,  $f(v)$  — цветом вершины  $v \in V$ , а граф  $G$ , каждой вершине которого сопоставляется какой-нибудь цвет, называется *цветным* либо *графом с цветными вершинами*. Такие графы обозначаются следующим образом:  $G = (V, \alpha, f)$ .

Аналогичным образом определяются графы с цветными ребрами. Далее для удобства граф с цветными вершинами будем называть *цветным графом* или просто *раскраской графа*, а граф с цветными ребрами — *реберной раскраской графа*.

*Изоморфизмом цветных графов*  $G_1 = (V_1, \alpha_1, f_1)$  и  $G_2 = (V_2, \alpha_2, f_2)$  называется изоморфизм графов  $G_1 = (V_1, \alpha_1)$  и  $G_2 = (V_2, \alpha_2)$ , сохраняющий цвета. Это биекция  $\varphi : V_1 \rightarrow V_2$ , при которой выполняются два условия:

- 1)  $\forall u, v \in V_1 : (u, v) \in \alpha_1 \Leftrightarrow (\varphi(u), \varphi(v)) \in \alpha_2$ ;
- 2)  $\forall v \in V_1 : f_1(v) = f_2(\varphi(v))$ .

Изоморфизм цветных графов также называется *цветным изоморфизмом*. Аналогично вводится понятие изоморфизма *графов с цветными ребрами*. Можно рассматривать вариант сильного изоморфизма, т.е. изоморфизма с точностью до цветов. Например, на рисунке показана 3-вершинная цепь  $P_3$  и 3 ее неизоморфные раскраски в 2 цвета. Первые две раскраски изоморфны с точностью до цветов.



Рис. 3-вершинная цепь  $P_3$  и ее раскраски в 2 цвета  
Fig. The path  $P_3$  and its 2-colorings

*Цветной автоморфизм графа* — это изоморфизм цветного графа на себя. Множество всех цветных автоморфизмов, включая и тождественный, образует



*группу автоморфизмов графа.* Две вершины графа называются *подобными*, если существует автоморфизм, отображающий одну вершину на другую. Множество подобных вершин называется *орбитой*.

Пусть  $S$  — множество, а  $F = \{S_1, \dots, S_p\}$  — семейство его различных непустых подмножеств, объединение которых дает  $S$ . *Граф пересечений*  $\Omega(F)$  семейства  $F$  определяется множеством  $V(\Omega(F)) = F$  и условием « $S_i$  и  $S_j$  смежны тогда и только тогда, когда  $i \neq j \wedge S_i \cup S_j \neq \emptyset$ » [11].

Рассмотрим множество  $X$  всех ребер графа  $G$  как семейство двухвершинных подмножеств множества вершин  $V(G)$ . *Реберным графом*  $L(G)$  графа  $G$  называется граф пересечений  $\Omega(X)$ . Таким образом, вершинами графа  $L(G)$  являются ребра графа  $G$ , и две вершины графа  $L(G)$  смежны тогда и только тогда, когда смежны соответствующие им ребра графа [11].

## 2. Построение неизоморфных $k$ -раскрасок графа

Мы будем рассматривать следующую основную задачу. Для заданного графа  $G = (V, \alpha)$  с  $n$  вершинами и заданным количеством цветов  $k$ ,  $1 < k \leq |V|$ , необходимо построить все неизоморфные  $k$ -раскраски графа  $G$ , причем все цвета должны в раскраске присутствовать. Очевидно, что для  $k = 1$  задача интереса не представляет, так как единственным образом можно раскрасить все вершины в один цвет.

Для задачи построения всех неизоморфных раскрасок вершин или ребер заданного графа ровно в  $k$  цветов неизвестно эффективное решение. Существуют различные переборные подходы для решения данной задачи [12–14], наиболее результативными из которых являются методы с использованием техники исключения изоморфизма (isomorphism rejection), т. е. без непосредственной проверки на изоморфизм. Хороший обзор по этим техникам можно найти в работе [5]. В основе таких подходов лежит метод канонических представителей. Идея метода состоит в том, что выбирается некоторое правило, по которому из всех изоморфных графов один граф объявляется каноническим. Часто в качестве канонического выбирается представитель с наибольшим или наименьшим кодом. В общем виде метод можно описать следующим образом [5]:

- 1) определяется способ кодирования объектов;
- 2) среди всех кодов изоморфных объектов выбирается канонический код;
- 3) порождаются все возможные уникальные структуры вместе с их кодами;
- 4) порожденная структура принимается, если ее код канонический, в противном случае — исключается.

На практике порождение всех кодов требует больших вычислительных ресурсов, поэтому используются различные методы оптимизации перебора. В настоящей работе производится сравнительная характеристика двух наиболее распространенных методов генерации: метода МакКея [1] и метода Рида–Фараджева [5]. Оба подхода используют технику канонических представителей, однако каждый отличается своим уникальным способом отсекающих неподходящих вариантов раскраски. Рассмотрим далее оба метода.

В *методе МакКея* генерация структур происходит итерационно от меньших к большему: от родителя к потомку. При этом одна и та же структура может быть потомком от нескольких родителей. Необходимо оставлять структуру в том случае, если она получена из подходящего родителя. Чтобы реализовать порождение всех искомым структур без проверки на изоморфизм, в методе МакКея предлагается выполнить следующие шаги:

- 1) для каждой структуры определяется уникальный (каноничный) родитель и



способ, которым данная структура должна быть получена из него;

2) структура принимается, только если она генерируется выбранным способом из канонического родителя;

3) из каждого родителя все неизоморфные потомки должны генерироваться по одному разу.

На выходе получается множество всех неизоморфных раскрасок заданного графа.

В *методе Рида – Фараджева*, в отличие от предыдущего алгоритма, определяется не способ получения канонического предка и его канонического потомка, а задается способ кодирования графов для получения канонических представителей и определяется способ сокращения перебора: если некоторая часть кода построена, то оставшаяся часть кода строится только в том случае, если получившийся код может быть каноническим. Удалось разработать алгоритмы, реализующие оба метода [13, 14]. Далее мы приведем алгоритмы и сравним результаты их работы для графов двух видов: цепей и циклов.

### 3. Построение неизоморфных графов с цветными ребрами

Задача поиска реберных раскрасок может быть сведена к задаче генерации вершинных раскрасок через построение для заданного графа  $G$  реберного графа  $L(G)$ : предлагается свести задачу генерации реберных раскрасок графа  $G$  к задаче генерации вершинных раскрасок реберного графа  $L(G)$ . Опишем способ приведения исходного графа к реберному.

Если графы  $G_1$  и  $G_2$  изоморфны, то графы  $L(G_1)$  и  $L(G_2)$  тоже изоморфны. Уитни установил [11], что обратное справедливо почти всегда, и указал при этом единственную пару различных графов  $K_3$  и  $K_{(1,3)}$ , имеющих один и тот же реберный граф.

Необходимо ввести алгоритм построения реберного графа для дальнейшего использования в алгоритме генерации раскрасок.

Пусть дан исходный граф  $G = (V, \alpha)$ .

1. Создается пустой граф  $L(G)$  без ребер и вершин.
2. Для каждого ребра  $(u, v) \in \alpha$  в граф  $L(G)$  добавляется новая вершина с меткой  $(u, v)$ .
3. Для каждой вершины  $(u, v) \in V(L(G))$  перебираются вершины  $(u', v')$  из  $L(V) \setminus \{(u, v)\}$ . Для каждой пары вершин сравниваются метки, и в зависимости от количества одинаковых элементов в реберный граф добавляется одно либо два ребра.
4. На выходе получается реберный граф  $L(G) = (V(L(G)), \alpha(L(G)))$ .

Теперь, построив граф  $L(G)$  и рассмотрев частные случаи, описанные Уитни, можно построить неизоморфные реберные раскраски для исходного графа  $G$ .

### 4. Построение неизоморфных раскрасок методом МакКея

Алгоритмизируем метод МакКея, применив его для построения неизоморфных раскрасок графа. В качестве вспомогательного инструмента для построения разбиений и вычисления орбит используется программный пакет *nauty* [2], реализованный МакКеем [1].

Алгоритм выглядит следующим образом.

На вход подается граф  $G = (V, \alpha)$ , количество цветов  $k : 1 < k \leq |V|$ .

1. При помощи программы *nauty* строится каноническая нумерация *lab* графа, вычисляются орбиты  $orb_0$ .





2. Создается вектор раскраски  $col$ , в него добавляется цвет 1 для первой вершины;  $colsize = 1$  — размер  $col$ ;  $mc = 1$  — максимальный по значению цвет, находящийся в  $col$ ;  $colcnt = 0$  — количество найденных раскрасок.

3. Перебираются все цвета от 1 до  $mc + 1$ , очередной цвет присваивается  $cur$ .

4. Для  $cur$  перебираются элементы раскраски  $col_i = \overline{1 \dots colsize}$ . Пусть  $col[i] > cur$  и  $orb_r[i] = orb_r[colsize + 1]$ , где  $orb_r$  — орбиты, полученные разбиением вершин раскраской  $r = (c_1, \dots, c_{i-1})$  — первые  $(i - 1)$  цвета  $col$ . Тогда данная раскраска объявляется изоморфной, отсекается и  $cur$  присваивается следующему цвету.

5.  $cur$  добавляется в  $col$ .

6. Вычисляется множество орбит  $orb_{col}$ , полученных разбиением вершин графа раскраской  $col$ .

7. Если  $cur > mc$  и  $mc \neq k$ , то  $mc = cur$ .

8. Если  $colsize < |V|$ , то  $colsize = colsize + 1$ , и производится переход на шаг 3.

9. Если  $colsize = |V|$  и  $mc = k$ , то производится дополнительная проверка раскраски: для каждой вершины  $i$  проверяется выполнение условия

$$\left( \forall j = \overline{1 \dots (i - 1)} \right) \left( col_j > col_i \wedge orb_{col_{j-1}}(j) \neq orb_{col_{j-1}}(i) \right).$$

Если условие выполняется, то раскраска считается изоморфной и не включается в ответ. Иначе необходимо вывести новую раскраску,  $colcnt = colcnt + 1$ , и завершить перебор для данной раскраски.

В 9-м пункте производится дополнительная проверка раскраски с учетом изоморфизма цветов. Если же изоморфизм раскрасок рассматривается как изоморфизм с точностью до цветов, то 9-й пункт алгоритма необходимо опустить.

Полученные раскраски удовлетворяют условию лексикографической минимальности и неизоморфности. Более того, не используется проверка на изоморфизм, только проверка на изоморфизм раскрасок, представленных в виде числовых векторов, что улучшает его эффективность.

## 5. Построение неизоморфных раскрасок методом Рида – Фараджева

Произведем алгоритмизацию метода Рида – Фараджева для построения неизоморфных раскрасок заданного графа. Сначала опишем общую схему поиска раскрасок, а затем дадим основной алгоритм генерации.

Схема поиска базируется на принципе перебора с отсекающими. На каждой итерации вычисляется множество орбит для заданной раскраски, выбирается по одному представителю из каждой орбиты, и каждый представитель раскрашивается во все цвета выбранным способом. Из набора полученных раскрасок отсекаются неканоничные.

Данная схема предполагает следующий ход генерации: на вход подается вектор цветов (раскраска) вершин графа, инициализированный первым цветом 1. Строится множество орбит, из каждой орбиты выбирается один наибольший ее представитель в виде индекса элемента раскраски. Каждый представитель раскрашивается во все цвета от 2 до  $k$ . Все раскраски, полученные данным образом, проверяются перекрашиванием — если перекрашенная раскраска меньше проверяемой, то проверяемая раскраска отсекается. Далее алгоритм генерации для всех неотсеченных раскрасок строит множество орбит и повторяет предыдущие действия, но с той оговоркой, что выбираются только те представители орбит, чей индекс меньше раскрашенного на предыдущем шаге элемента раскраски. Например, если для раскраски размерности



пять предыдущим раскрашиваемым элементом был четвертый элемент, то на текущем шаге генерации будут раскрашены только те представители орбит, чей индекс меньше четырех. Генерация заканчивается, когда на очередном шаге не останется неотсеченных раскрасок.

Оформим данную схему в алгоритм генерации неизоморфных вершинных  $k$ -раскрасок без проверки на изоморфизм методом Рида – Фараджева.

Для вычисления орбит по заданному разбиению будет использоваться программа *nauty* [2], реализованная по разработанному Бренданом МакКеем алгоритму [1].

На вход подается граф  $G = (V, \alpha)$ , количество цветов  $k : 1 < k \leq |V|$ .

1. Задается последовательность  $clr = (1, \dots, 1)$ ,  $|clr| = |V|$  — инициализирующая последовательность цветов заданного графа, где  $clr_i$  является цветом  $i$ -й вершины.

2. По заданной  $clr$  вычисляются орбиты при помощи программы *nauty*.

3. Из каждой орбиты берется по одному наибольшему представителю  $o$ .

4. Каждый выбранный представитель  $clr_o$  последовательно раскрашивается в цвета от 2 до  $k$ , получая раскраски с раскрашенными вершинами  $clr_o$ .

5. Каждая  $clr$ , полученная перекрашиванием  $o$ -й вершины в какой-нибудь цвет, проверяется на каноничность ее родителя путем перекрашивания по следующим правилам. Раскраска перекрашивается следующим образом: инициализируем переменную  $l$  первым цветом; циклически проходимся по всем вершинам; если текущая вершина ранее не встречалась в раскраске, присваиваем ей цвет  $l$  и увеличиваем  $l$  на единицу; если же вершина уже присутствовала в раскраске, раскрашиваем ее в присвоенный ей цвет. Если перекрашенная раскраска получилась лексикографически меньше проверяемой, то проверяемая раскраска отсекается.

6. Если не осталось ни одной неотсеченной раскраски, то алгоритм заканчивается.

7. Для каждой неотсеченной раскраски снова строится множество орбит, из каждой орбиты выбирается по одному наибольшему представителю  $o$  с той оговоркой, что индекс представителя должен быть меньше индекса раскрашенной на предыдущем шаге генерации вершины, производится переход на шаг 4.

Полученные раскраски удовлетворяют условию лексикографической минимальности и неизоморфности. Проверка на изоморфизм не используется.

## 6. Сравнение алгоритмов генерации неизоморфных раскрасок методами МакКея и Рида – Фараджева

Описанные выше алгоритмы были реализованы на языке программирования Python. Программа использует однопоточный режим исполнения, однако может быть легко перенесена на случай многопоточной генерации раскрасок. Для того чтобы запускать алгоритмы в многопоточном режиме, необходимо для каждого уровня (итерации) генерации объектов запускать новый поток, который будет завершен при завершении нового набора неизоморфных структур.

Проведем сравнение методов генерации, основываясь на времени выполнения реализованных алгоритмов. В данной работе для сравнения были выбраны два класса графов: цепи и циклы. Напомним, что *циклом*  $C_n$  называется граф  $G = (V, \alpha)$ , где  $V = \{v_1, v_2, \dots, v_n\}$ , и  $\alpha = \{(v_i, v_j) : |i - j| = 1\} \cup \{(v_1, v_n), (v_n, v_1)\}$ . *Цепью*  $P_n$  называется граф  $G = (V, \alpha)$ , где  $V = \{v_1, v_2, \dots, v_n\}$ , и  $\alpha = \{(v_i, v_j) : |i - j| = 1\}$

Сравним время генерации раскрасок для всех цепей и циклов с числом вершин  $n$  от 3 до 9. Результаты генерации раскрасок для цепей с  $n$  вершинами представлены в табл. 1, а для циклов — в табл. 2. Время работы программы представлено в секундах.



Таблица 1 / Table 1

Сравнение времени работы методов МакКея и Рида – Фараджева на цепях  
The comparison of the running time of the McKay and Reed – Faradzhev methods on paths

$n$	$k$	Количество раскрасок Number of colorings	Метод МакКея McKay's method	Метод Рида – Фараджева Reed – Faradzhev's method
3	2	3	0.0002	0.0002
4	2	6	0.0004	0.0003
	3	6	0.0023	0.0014
5	2	13	0.0008	0.0006
	3	24	0.0082	0.0042
	4	10	0.0475	0.0173
6	2	27	0.0020	0.0012
	3	88	0.0313	0.0142
	4	65	0.2370	0.0808
	5	15	1.1499	0.3123
7	2	5	0.0040	0.0026
	3	291	0.1098	0.0457
	4	348	1.1181	0.36
	5	140	6.7178	1.8045
	6	21	29.6244	6.4205
8	2	115	0.0118	0.0057
	3	951	0.4270	0.1528
	4	1698	5.3225	1.6220
	5	1050	38.8733	9.8318
	6	266	203.6058	41.1898
	7	28	795.6863	146.3406
9	2	231	0.0255	0.0125
	3	3199	1.3646	0.5251
	4	10942	23.9317	7.1255
	5	17890	222.7841	54.4089
	6	20536	1462.2332	287.4058
	7	20998	6415.6784	1002.7468
	8	36	25239.4394	3526.7013

Таблица 2 / Table 2

Сравнение времени работы методов МакКея и Рида – Фараджева на циклах  
The comparison of the running time of the McKay and Reed – Faradzhev methods on cycles

$n$	$k$	Количество раскрасок Number of colorings	Метод МакКея McKay's method	Метод Рида – Фараджева Reed – Faradzhev's method
3	2	2	0.0002	0.0002
4	2	4	0.0002	0.0002
	3	9	0.0017	0.0009
5	2	7	0.0004	0.0003
	3	25	0.0059	0.0033
	4	34	0.0372	0.0152





Окончание табл. 2 / End of Table 2

$n$	$k$	Количество раскрасок Number of colorings	Метод МакКея McKay's method	Метод Рида – Фараджева Reed – Faradzhev's method
6	2	13	0.0012	0.0011
	3	73	0.0200	0.0103
	4	127	0.1825	0.0664
	5	141	1.1719	0.2687
7	2	24	0.0019	0.0013
	3	219	0.0945	0.0325
	4	500	0.9011	0.3196
	5	625	6.4083	1.8937
	6	645	27.7829	6.8833
8	2	46	0.0029	0.0025
	3	655	0.2493	0.0930
	4	1978	4.3110	1.4072
	5	2885	35.7415	9.4039
	6	3130	186.3902	37.2708
	7	3157	775.4324	124.2052
9	2	93	0.0121	0.0054
	3	2027	0.9505	0.3762
	4	8024	21.1162	6.2137
	5	13916	187.8341	43.5724
	6	16296	1154.5761	235.9630
	7	434	4884.8862	892.7705
	8	35	18988.9581	3017.8236

Можно заметить, что алгоритм, построенный на основе метода Рида – Фараджева, работает в несколько раз быстрее, чем алгоритм на основе метода МакКея на тех же входных данных.

### Заключение

В данной работе было произведено исследование генерации неизоморфных графов с цветными вершинами и цветными ребрами, т. е. генерация вершинных и реберных  $k$ -раскрасок соответственно. Была использована техника генерации раскрасок без проверки на изоморфизм. Описываются два алгоритма генерации неизоморфных раскрасок методами МакКея и Рида – Фараджева. Произведена сравнительная характеристика двух методов на цепях и циклах. Из полученных результатов можно сделать вывод, что для цепей и циклов алгоритм, построенный на основе метода Рида – Фараджева, работает быстрее.

### Список литературы

1. McKay B. D., Piperno A. Practical graph isomorphism, II // Journal of Symbolic Computation. 2013. Vol. 60. P. 94–112. <https://doi.org/10.1016/j.jsc.2013.09.003>
2. McKay B. D., Piperno A. Nauty and Traces: Graph canonical labeling and automorphism group computation. URL: <https://pallini.di.uniroma1.it/> (дата обращения: 01.05.2020).



3. Meringer M. Fast generation of regular graphs and construction of cages // *Journal of Graph Theory*. 1999. Vol. 30. P. 137–146. [https://doi.org/10.1002/\(SICI\)1097-0118\(199902\)30:2<137::AID-JGT7>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0118(199902)30:2<137::AID-JGT7>3.0.CO;2-G)
4. Brinkmann G., Goedgebeur J., McKay B. D. Generation of cubic graphs // *Discrete Mathematics and Theoretical Computer Science, DMTCS*. 2011. Vol. 13, no. 2. P. 69–79.
5. Brinkmann G. Isomorphism rejection in structure generation programs // *Discrete Mathematical Chemistry* / eds. P. Hansen, P. W. Fowler and M. Zheng. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 2000. Vol. 51. P. 25–38.
6. Hayes J. P. A graph model for fault-tolerant computing system // *IEEE Transactions on Computers*. 1976. Vol. C-25, no. 9. P. 875–884. <https://doi.org/10.1109/TC.1976.1674712>
7. Jensen T., Toft B. *Graph Coloring Problems*. Wiley-Interscience, 1994. 320 p.
8. *Topics in Chromatic Graph Theory* / ed. by L. W. Beineke, R. J. Wilson. Cambridge : Cambridge University Press, 2015. 370 p. (Encyclopedia of Mathematics and its Applications 156). <https://doi.org/10.1017/CBO9781139519793>
9. Lewis R. M. R. *A Guide to Graph Colouring. Algorithms and Applications*. Springer, Cham, 2016. 253 p. <https://doi.org/10.1007/978-3-319-25730-3>
10. Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. Москва : Наука, 1997. 368 с.
11. Харари Ф. Теория графов. Москва : Мир, 1973. 296 с.
12. Абросимов М. Б., Разумовский П. В. О генерации неизоморфных вершинных  $k$ -раскрасок // *Прикладная дискретная математика. Приложение*. 2017. № 10. С. 136–138. <https://doi.org/10.17223/2226308X/10/53>
13. Разумовский П. В., Абросимов М. Б. О генерации неизоморфных  $k$ -раскрасок методом МакКея // *Компьютерные науки и информационные технологии : материалы междунар. науч. конф.* Саратов : ИЦ «Наука», 2018. С. 318–320.
14. Абросимов М. Б., Разумовский П. В. О генерации неизоморфных раскрасок методом Рида – Фараджева // *Прикладная дискретная математика. Приложение*. 2019. № 12. С. 173–176. <https://doi.org/10.17223/2226308X/12/48>

### References

1. McKay B. D., Piperno A. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 2013, vol. 60, pp. 94–112. <https://doi.org/10.1016/j.jsc.2013.09.003>
2. McKay B. D., Piperno A. *Nauty and Traces: Graph canonical labeling and automorphism group computation*. Available at: <https://pallini.di.uniroma1.it/> (accessed 1 May 2020).
3. Meringer M. Fast generation of regular graphs and construction of cages. *Journal of Graph Theory*, 1999, vol. 30, pp. 137–146. [https://doi.org/10.1002/\(SICI\)1097-0118\(199902\)30:2<137::AID-JGT7>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1097-0118(199902)30:2<137::AID-JGT7>3.0.CO;2-G)
4. Brinkmann G., Goedgebeur J., McKay B. D. Generation of cubic graphs. *Discrete Mathematics and Theoretical Computer Science. DMTCS*, 2011, vol. 13, no. 2, pp. 69–79.
5. Brinkmann G. Isomorphism rejection in structure generation programs. In: P. Hansen, P. W. Fowler, M. Zheng, eds. *Discrete Mathematical Chemistry*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2000, vol. 51, pp. 25–38.
6. Hayes J. P. A graph model for fault-tolerant computing system. *IEEE Transactions on Computers*, 1976, vol. C-25, no. 9, pp. 875–884. <https://doi.org/10.1109/TC.1976.1674712>
7. Jensen T., Toft B. *Graph Coloring Problems*. Wiley-Interscience, 1994. 320 p.
8. L. W. Beineke, R. J. Wilson, eds. *Topics in Chromatic Graph Theory*. Cambridge, Cambridge University Press, 2015. 370 p. (Encyclopedia of Mathematics and its Applications 156). <https://doi.org/10.1017/CBO9781139519793>
9. Lewis R. M. R. *A Guide to Graph Colouring. Algorithms and Applications*. Springer, Cham, 2016. 253 p.



10. Bogomolov A. M., Salii V. N. *Algebraicheskie osnovy teorii diskretnykh sistem* [Algebraic Foundations of the Theory of Discrete Systems]. Moscow, Nauka, 1997. 368 p (in Russian).
11. Harary F. *Graph Theory*. Addison-Wesley, 1969. 275 p. (Russ. ed.: Moscow, Mir, 1973. 296 p.).
12. Abrosimov M. B., Razumovsky P. V. About generation of non-isomorphic vertex  $k$ -colorings. *Prikladnaya Diskretnaya Matematika. Supplement*, 2017, no. 10, pp. 136–138 (in Russian). <https://doi.org/10.17223/2226308X/10/53>
13. Razumovsky P. V., Abrosimov M. B. About generation of non-isomorphic  $k$ -colorings by McKay's method. In: *Komp'iuternye nauki i informatsionnye tekhnologii* [Computer Science and Information Technologies: Proceedings of the International Scientific Conference]. Saratov, Publishing Center "Nauka", 2018, pp. 318–320 (in Russian).
14. Abrosimov M. B., Razumovsky P. V. About non-isomorphic graph colouring generating by Read – Faradzhev method. *Applied Discrete Mathematics. Supplement*, 2019, no. 12, pp. 173–176 (in Russian). <https://doi.org/10.17223/2226308X/12/48>

Поступила в редакцию / Received 10.06.2020

Принята к публикации / Accepted 12.10.2020

Опубликована / Published 31.05.2021