

УЧРЕЖДЕНИЕ РОССИЙСКОЙ АКАДЕМИИ НАУК
САНКТ-ПЕТЕРБУРГСКОЕ ОТДЕЛЕНИЕ
МАТЕМАТИЧЕСКОГО ИНСТИТУТА ИМ. В. А. СТЕКЛОВА РАН

На правах рукописи

Гравин Николай Вадимович

Некоторые аспекты правильных раскрасок графов

01.01.09 — дискретная математика и математическая кибернетика

Диссертация

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:

доцент, к.ф.-м.н. Д. В. Карпов

Санкт-Петербург

2010

Оглавление

1	Введение	4
1.1	Определения	6
1.1.1	Базовые понятия	6
1.1.2	Понятия связанные с раскрасками	8
2	Гиперграфы и Динамические раскраски	10
2.1	Верхние оценки на хроматическое число для гипер-графов	11
2.2	Верхние оценки на динамическое хроматическое число . .	13
3	Теорема Брукса и Динамические Раскраски	15
3.1	(c, p) -невыврожденность	15
3.2	Доказательство Основного Результата	19
3.2.1	Часть первая.	22
3.2.2	Часть вторая.	30
4	Алгоритмы для Раскрасок Графов	50
4.1	Обозначения	51
4.1.1	Терминология	52
4.1.2	Основной результат	53
4.2	Задача Списковой d -раскраски	55

4.2.1	Алгоритм	56
4.2.2	Реализация процедур	59
4.2.3	Доказательство корректности алгоритма	62
4.2.4	Анализ сложности алгоритма	64
4.3	Конструктивное доказательство теоремы Брукса	67

Глава 1

Введение

Graph theory is very important for mathematics and informatics

Теория графов является одним из важнейших и интереснейших разделов математики. Помимо многочисленных приложений в различные области математики таких, как теория вероятностей, комбинаторика, алгебра и топология, она является краеугольным камнем в современной информатике, как теоретической, так и прикладной. В любом учебнике по алгоритмам едва ли не треть всего материала будет посвящена алгоритмам на графах. Терминология теоретической информатики же не мыслима без использования понятий теории графов как то дерево, вершина, ребро, путь, цикл и более сложных, таких как экспандер.

Colorings are one of the most difficult though very popular and long studied topic

History and overview of main results in coloring Правильные раскраски графов на сегодняшний день вероятно являются одной из самых популярных тем в теории графов. Хотя уже проблема раскрашиваемо-

сти графа в 3 цвета является NP-полной задачей, существует несколько положительных результатов описывающих случаи, где число допустимых цветов близко к Δ — максимальной степени графа. Пожалуй самой известной среди них является теорема Брукса [14], утверждающая, что связный граф, не являющийся нечетным циклом или кликой, можно всегда раскрасить в Δ цветов. Однако, вопрос становится гораздо труднее уже для раскраски в $\Delta - 1$ цвет. В 1977 году Бородин и Косточка [13] выдвинули гипотезу, что для $\Delta \geq 9$ графы без клик размера Δ могут быть раскрашены в $(\Delta - 1)$ цвет. Гипотеза до сих пор остается открытой, однако в случае $\Delta \geq 10^{14}$ она была положительно решена Риидом [24] при помощи вероятностного метода.

Другое обобщение теоремы Брукса связанное со списковыми раскрасками было независимо предложено Визингом [28] и Ердешем с соавторами [16]. Недавно это направление даже приобрело большую популярность, чем правильные раскраски, см. например [18, 27, 22, 20] и многие другие источники, один из самых последних результатов [19]. В 1994 на конференции по теории графов в Обервольхе Томассеном было отмечено, что помимо теоремы Брукса можно также обобщить для списковых раскрасок классическую теорему Галлаи [17], в которой описываются все подграфы k -раскрасочно-критических графов индуцированных на вершинах степени $k - 1$. Удивительно, но как отметил Косточка в [21], обобщение теоремы Галлаи может быть легко выведено из результатов Бородина [11, 12] и Ердеша, Рубина и Тейлора [16]. Более подробно, Ердеш характеризовал d -списочно раскрашиваемые графы, как графы содержащие блок, который не является ни кликой, ни нечетным циклом. В этой характеристике никак не упоминается случай раскраски

не d -списочно раскрашиваемого графа, если помимо самого графа задан набор списков (в этом случае ответ может быть отрицательным). Последнее было сделано Бородиным в малоизвестной работе [11].

My contribution

organization of the dissertation

1.1 Определения

В диссертации, если не оговорено противное, мы будем рассматривать неориентированные графы без петель и кратных ребер, множество вершин графа G мы всегда будем обозначать через $V(G)$, а множество ребер — через $E(G)$.

В дальнейшем нам понадобятся несколько стандартных определений, которые мы приводим ниже.

1.1.1 Базовые понятия

Определение. Пусть $S \subseteq V(G)$, индуцированным подграфом $G(S)$ графа G называется граф на множестве вершин S содержащий те и только те ребра G , оба конца которых лежат в S .

Пусть G — произвольный граф. Для $W \subset V(G)$ через $G - W$ мы будем обозначать граф, получающийся при удалении всех вершин множества W и всех выходящих из них ребер. Для $F \subseteq E(G)$ через $G - F$ мы будем обозначать граф, который получается при удалении из G всех ребер множества F (то есть, $G - F = (V(G), E(G) \setminus F)$).

Через $\deg_G(u)$ или $\deg(u)$ мы будем обозначать степень вершины u в G , т.е. число инцидентных с u ребер. Следуя книге [1] через $\delta(G)$ и $\Delta(G)$ мы будем обозначать минимальную и максимальную степени графа G . *Окрестностью* вершины $v \in V(G)$ в графе G называется множество $N_G(v) = \{u \in V(G) : uv \in E(G)\}$. Вершины из $N_G(v)$ называются соседями v .

Определение. Пусть $S \subseteq V(G)$, обозначим через $N_v(S)$ окрестность вершины v в графе $G(S \cup \{v\})$. Мы будем писать для краткости N_v в случае $S = V(G)$.

Определение. Обозначим через $|G|$ размер графа G , определяемый как количество ребер плюс количество вершин в G .

Граф называется *связным*, если между любыми двумя его вершинами существует путь. Множество вершин несвязного графа разбивается на *компоненты связности*. (Под компонентой связности мы понимаем максимальное по включению множество вершин графа, любые две из которых связаны путем.)

Точкой сочленения называется вершина, после удаления которой связный граф распадается на более чем одну компоненту связности. Граф называется *двусвязным*, если при удалении любой вершины, он остается связным (мы предполагаем здесь, что пустой граф является связным). *Блоком* связного графа G называется максимальный по включению двусвязный подграф G .

Замечание. Все вышеописанные определения обобщаются для мультиграфов, где разрешены петли и кратные ребра.

Правильной раскраской графа G называется отображение $c : V(G) \rightarrow \mathbb{N}$ такое, что $c(v) \neq c(u)$, если $uv \in E(G)$. Раскраска c называется k -раскраской, если всего было использовано не более k цветов. Пусть c – правильная раскраска G в k цветов, а множество $V' \subseteq V(G)$, тогда через $c(V')$ мы будем обозначать ограничение отображения c на множество V' , т.е. правильную раскраску графа $G(V')$ в k цветов.

1.1.2 Понятия связанные с раскрасками

Естественно рассмотреть модель с более жесткими ограничениями, например где допустимые цвета предварительно заданы для каждой вершины отдельно.

Более формально, списковым набором является функция \mathcal{L} ставящая в соответствие каждой вершине $v \in V(G)$ некоторое множество $\mathcal{L}(v)$ натуральных чисел, которые зовутся *допустимыми цветами* вершины v . Тогда \mathcal{L} -раскраской G будет называться такая правильная раскраска c графа G , что каждой вершине v соответствует цвет $c(v) \in \mathcal{L}(v)$. В случае, если $|\mathcal{L}(v)| \geq k$ для любой вершины $v \in V(G)$, мы будем говорить, что \mathcal{L} является k -набором, если же $|\mathcal{L}(v)| \geq \deg(v)$ для каждой вершины $v \in V(G)$, мы будем говорить, что \mathcal{L} является d -набором. *Частичной \mathcal{L} -раскраской* подмножества вершин S графа G называется \mathcal{L} -раскраска индуцированного подграфа $G(S)$. \mathcal{L} -раскраска \mathcal{C} графа G называется *расширением* частичной раскраски \mathcal{C}' на множестве вершин S , если $\mathcal{C}(v) = \mathcal{C}'(v)$ для всех $v \in S$.

Понятие правильной раскраски можно несколькими способами обобщить на гиперграфы. Мы будем рассматривать наиболее естественный вариант, предложенный Полем Ердешем, в котором каждое гипер ребро

должно содержать по крайней мере две вершины разного цвета.

Глава 2

Гиперграфы и Динамические раскраски

Определение 2.1. Назовем k -редукцией графа G максимальный по включению подграф $R_k(G)$ графа G , в котором степень любой вершины по крайней мере k . Если такого подграфа нет, то определим пустой граф как k -редукцию G . В последнем случае G будет называться k -редуцируемым графом.

Замечание. k -редукция G определена единственным образом, так как индуцированный подграф объединения двух подграфов G с минимальной степенью по крайней мере k , также будет иметь минимальную степень по крайней мере k .

2.1 Верхние оценки на хроматическое число для гипер-графов

Теорема 2.1. Пусть $\mathcal{H} = (V, \mathcal{E})$ является δ -гиперграфом с максимальной степенью вершины Δ . Тогда \mathcal{H} можно раскрасить в $\lceil \frac{2\Delta}{\delta} + 1 \rceil$ цветов.

Доказательство. Обозначим $\lceil \frac{2\Delta}{\delta} + 1 \rceil$ через k .

Определение 2.2. Назовем **графом образом** или просто **образом** некоторого гиперграфа $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ любой граф с мульти-ребрами $G = (V, E)$, для которого $V = \mathcal{V}$ и между ребрами E и \mathcal{E} есть биективное соответствие такое, что каждое гипер-ребро $\mathcal{F} \in \mathcal{E}$ включает соответствующее ребро из E .

Чтобы доказать теорему, достаточно построить раскрашиваемый в k цветов образ гипер-графа \mathcal{H} . Предположим от противного, что такого образа не существует, рассмотрим тогда образ G_1 графа \mathcal{H} с наименьшим возможным $|R_k(G_1)|$. Так как каждый k -редуцируемый граф k -раскрашиваем, $R_k(G_1)$ не пусто. Зафиксируем граф $H_1 = R_k(G_1)$ и обозначим множество его вершин через S_1 .

Для доказательства теоремы нам понадобится конструкция. В ней на шаге i мы строим следующее:

1. последовательность множеств $S_1 \subset S_2 \subset \dots \subset S_i \subseteq V$;
2. последовательность графов H_j , где $j \in [1, i]$ и $V(H_j) = S_j$;
3. существует по крайней мере один образ G_i графа \mathcal{H} с $R_k(G_i) = H_1$ и $H_j = G_i(S_j)$ для всех $j \in [1, i]$; обозначим множество всех таких

образов как \mathcal{G}_i ;

4. для любого $j \in [1, i]$ не существует образа G гипер графа \mathcal{H} с $R_k(G) = H_1$ и $E(G(S_j)) \supset E(H_j)$, т.е. H_j является минимальным по включению индуцированным подграфом на множестве S_j взятом среди всех образов G гипер графа \mathcal{H} с условием $R_k(G) = H_1$;
5. H_j имеет больше чем $\frac{\Delta}{\delta}|S_j|$ ребер.

Легко видеть, что S_1 , H_1 и G_1 удовлетворяют всем условиям для $i = 1$.

Так как $|S_i|$ ограничено сверху, для завершения доказательства нам достаточно продолжить конструкцию на следующий $i+1$ шаг для любого i .

1. Заметим, что обязательно найдется ребро $e \in E(H_i)$ и соответствующее ему гипер-ребро $\mathcal{F} \in \mathcal{E}$ с вершиной $v \in \mathcal{F}$, $v \notin S_i$.

Последнее выполнено, так как в противном случае по условию 5 для $j = i$ в множестве S_i будет лежать больше чем $\frac{\Delta}{\delta}|S_j|$ гипер-ребер графа \mathcal{H} размера δ . Следовательно, по принципу Дирихле найдется вершина \mathcal{H} степени больше чем Δ , что противоречит условию теоремы.

2. Пусть $e = (u, w)$. Для каждого графа $G \in \mathcal{G}_i$ попытаемся заменить e на (u, v) . Получившийся граф G_e будет образом \mathcal{H} , $E(G_e(S_j)) \subseteq E(H_j)$ для всех $j \in [1, i]$ и кроме того для некоторого j включение будет строгим. Таким образом по условию 4 нашей конструкции получаем, что $R_k(G_e) \neq H_1$. Если $v \notin R_k(G_e)$, тогда $R_k(G_e) \subseteq R_k(G)$. Но

значение $|R_k(G)|$ было минимально возможным и значит мы приходим к противоречию. Следовательно $v \in R_k(G_e)$. Для каждого графа $G \in \mathcal{G}_i$ положим $S_G = V(R_k(G_e))$ и $H_G = G(S_i \cup S)$. Таким образом получаем $v \in S_G$ и значит $v \in V(H_G)$.

3. Возьмем G из \mathcal{G}_i с минимально возможным $|H_G|$ и положим $G_{i+1} = G$. Возьмем в качестве S_{i+1} множество $S_G \cup S_i$, а в качестве H_{i+1} возьмем $G(S_{i+1})$. Так как $v \in S_{i+1}$ и $v \notin S_i$, имеем $S_i \subset S_{i+1} \subseteq V$. Отсюда получаем, что условие 1 конструкции выполнено для $i + 1$; свойства 2 и 3 выполняются по определению H_{i+1} и S_{i+1} ; условие 4 выполнено в силу выбора $H_G = H_{i+1}$; каждая вершина из S_G в H_G имеет степень по крайней мере $\frac{2\Delta}{\delta}$ и $H_G(S_i) = H_i$ имеет больше чем $\frac{\Delta}{\delta}|S_i|$ ребер, следовательно в H_G имеется больше чем, $\frac{\Delta}{\delta}|S_i| + \frac{|S_G \setminus S_i| 2\Delta}{2\delta} = \frac{\Delta}{\delta}|S_{i+1}|$ ребер; таким образом свойство 5 тоже выполнено для $i + 1$, что завершает доказательство теоремы.

□

2.2 Верхние оценки на динамическое хроматическое число

Определение 2.3. Динамической раскраской графа G в k цветов называется отображение $c : V(G) \rightarrow \{1, 2, \dots, k\}$ такое, что для любой вершины v графа G в множестве $N_v(G)$ найдется по крайней мере два различных цвета.

Замечание. Динамическая раскраска не обязана быть правильной раскраской.

Теорема 2.2. *Дан граф G с максимальной степенью Δ и минимальной степенью δ . Тогда G допускает динамическую раскраску в $\lceil \frac{2\Delta}{\delta} + 1 \rceil$ цветов.*

Доказательство. Рассмотрим следующий гипер-граф \mathcal{H} .

- Множество вершин \mathcal{H} совпадает с $V(G)$.
- Множество гипер-ребер \mathcal{H} состоит из множеств N_v для всех $v \in V(G)$.

Теорема эквивалентна тому факту, что \mathcal{H} допускает k -раскраску. Каждое гипер-ребро \mathcal{H} имеет размер по крайней мере δ и каждая вершина \mathcal{H} лежит не более чем в Δ гипер-ребрах. Настоящая теорема легко следует из теоремы 2.1, если мы произвольным образом уменьшим размер каждого гипер ребра до δ .

□

Глава 3

Теорема Брукса и Динамические Раскраски

Настоящая глава посвящена обобщению широко известной теоремы Брукса [2], которая говорит нам, что $\chi(G) \leq \Delta(G)$ для любого графа G без полных подграфов на $d + 1$ вершине с $\Delta(G) > 2$.

Обобщение заключается в том, что мы дополнительно накладываем требование (c, p) -невырожденности на раскраску графа.

3.1 (c, p) -невырожденность

Ранее предпринимались попытки получить похожие на теорему Брукса утверждения с дополнительным условием *c-невырожденности* говорящем, что среди соседей любой вершины графа степени хотя бы c должны найтись c вершин различных цветов. Так, в работах [3, 4] были найдены оценки на количество цветов, в которое можно правильным 2 -невырожденным образом покрасить граф. Случай $c = 2$ особенно ин-

интересен, в виду тесной связи динамических раскрасок и правильных раскрасок гиперграфов. Стоит отметить однако, что полученные в этих работах результаты не являются обобщениями теоремы Брукса, т.к. количество цветов, в которое красится граф, больше Δ .

Дальнейшие исследования, для произвольного значения c , можно найти в работах [5, 6]. В них авторы замечают, что было бы интересно получить обобщение теоремы Брукса в направлении невырожденности раскраски. Несмотря на значительный интерес к данной теме, задача подобного обобщения до сих пор не решена и представляется слишком сложной, чтобы решать ее в подобной формулировке, т.е. когда условие невырожденности накладывается на *все* вершины графа степени хотя бы c .

Приведем соображение, заставляющее задуматься о необходимости изменения постановки задачи. Действительно, рассмотрим двудольный граф, у которого все вершины первой доли имеют степень c , а второй доли Δ . Условие невырожденности для любой вершины v из первой доли говорит нам, что соседи v покрашены в различные цвета, что эквивалентно наличию клики при правильной раскраске. Условие невырожденности можно тогда переформулировать в терминах нахождения правильной раскраски вершин второй доли, в которой мы построили клики на всевозможных окрестностях вершин первой доли. Необходимо отметить, что в новом графе вершины второй доли могут иметь степень порядка $(c - 1)\Delta$. Таким образом даже в этом частном случае при $c > 2$ трудно надеяться на нахождение правильной невырожденной раскраски с количеством цветов близким к Δ .

Ключевым соображением, позволившим сделать серьезные продви-

жения в направлении обобщения теоремы Брукса является то, что мы потребуем несколько иное условие невырожденности раскраски, которое назовем (c, p) -невырожденностью. А именно, чтобы раскраска была невырожденной в обычном смысле не для всех вершин, а для вершин *достаточно большой степени*.

Определение 3.1. Пусть даны натуральные числа $c \geq 2$ и $p \geq c$. Раскраска графа G называется (c, p) -невырожденной, если для любой вершины G степени хотя бы p среди ее соседей найдутся вершины хотя бы c различных цветов.

Изменение условия позволяет нам сформулировать и доказать следующую весьма общую теорему, а также получить в процессе доказательства несколько интересных следствий.

Теорема 3.1. Пусть дано число $D \geq 3$ и граф G без клик на $D + 1$ и $\Delta(G) \leq D$. Тогда для любого $c \geq 2$ существует правильная (c, p) -невырожденная раскраска вершин графа G в D цветов, где

$$p = (c^3 + 8c^2 + 19c + 6)(c + 1).$$

В связи с нашим результатом нельзя не упомянуть работу [7], где авторами получены следующие довольно сильные результаты.

Определение 3.2. Раскраска графа называется β -frugal, если никакая вершина графа не имеет в своей окрестности более чем β соседей одного цвета.

Теорема 3.2. У Любого графа G с максимальной степенью $\Delta \geq \Delta_0 = e^{10^7}$ существует правильная $\lceil \log^8 \Delta \rceil$ -frugal раскраска в $(\Delta + 1)$ цвет.

Из этой теоремы очевидным образом можно получить:

Следствие. У Любого графа G с максимальной степенью $\Delta \geq \Delta_0 = e^{10^7}$ существует правильная $(\frac{\Delta_0}{\lceil \log^8 \Delta_0 \rceil}, \Delta_0)$ -невырожденная раскраска в $(\Delta + 1)$ цвет.

Отсюда можно получить оценку невырожденности раскраски асимптотически гораздо лучшую, чем наша, используя при этом всего один лишний цвет. Однако между правильной раскраской в Δ цветов и $\Delta + 1$ цвет существует большая разница. Можно с легкостью рекурсивно покрасить граф в $\Delta(G) + 1$ цвет, однако аналогичное утверждение для $\Delta(G)$ цветов представляется совсем не очевидным. Также необходимо отметить, что это следствие как и все результаты в работе [7] получены только для очень больших графов. По сравнению с числами $\Delta \geq \Delta_0 = e^{10^7}$, оценка $p = (c^3 + 8c^2 + 19c + 6)(c + 1)$ представляется значительно меньшей. Она дает нам нетривиальный результат уже при $c = 2$ и $p = 252$, то есть уже для графов с $\Delta \geq 252$.

Существует интересное соображение касающееся β -frugal и (c, p) -невырожденных раскрасок. Почему мы вообще изучаем правильные раскраски графов? Одним из объяснений может служить то, что раскраска графа G в c цветов есть в точности гомоморфизм G в граф K_c (определение и более полная информация касающаяся гомоморфизмов графов можно найти например в книге [8]). Далее можно сделать весьма естественный шаг и рассмотреть для каждой вершины графа множество ее соседей, или ее окрестность. Тогда условие (c, p) -невырожденности будет в точности означать, что для каждой вершины v с $|N_G(v)| \geq p$, размер образа ее окрестности при гомоморфизме будет не меньше c . То есть (c, p) -невырожденность накладывает ограничение снизу на размер

образа отображения. А свойство того, что раскраска является β -frugal, будет устанавливать верхнюю оценку на размер “ядра” отображения, где под размером ядра естественно понимать максимальный размер подмножества окрестности отображающегося в один цвет.

3.2 Доказательство Основного Результата

Одним из основных этапов доказательства теоремы 1 является доказательство следующей теоремы 2, которая представляет отдельный интерес.

Теорема 3.3. *Пусть дан граф G без клик размера $D + 1$ с максимальной степенью не большей, чем D . И пусть кроме того $D = \sum_{i=1}^{c+1} \alpha_i$, где $\alpha_i \geq 2$. Тогда в множестве Ξ раскрасок графа G в $c + 1$ цвет найдется такая раскраска ξ , что:*

1) $\Phi(\xi) = \min_{\psi \in \Xi} \Phi(\psi)$, где $\Phi = \sum_{i=1}^{c+1} \frac{f_i}{\alpha_i}$, а f_i – это количество ребер графа G , соединяющих две вершины i -ого цвета,

2) для любого i раскраска ξ не содержит клик размера $\alpha_i + 1$ на вершинах i -ого цвета.

3) $\Delta_{G(V_i)} \leq \alpha_i$, где V_i это множество всех вершин i -ого цвета в G .

В частности, одним из прямых следствий этой теоремы является результат, близкий по смыслу к полученному л. Ловашем в работе [9], который говорит нам, что вершины любого графа G с $\Delta_G \leq k$ можно разбить на n множеств V_1, V_2, \dots, V_n таких, что $\Delta_{G(V_i)} \leq k_i$, где $\sum_{i=1}^n k_i = k - n + 1$.

Следствие. *Пусть дан граф G без клик размера $D + 1$ с максимальной степенью не большей, чем D . И пусть кроме того $D = \sum_{i=1}^k \alpha_i$, где*

$\alpha_i \geq 2$. Тогда вершины графа G можно разбить на k непересекающихся множеств V_1, V_2, \dots, V_k таких, что для любого $i \in \bar{k}$ в графе $G(V_i)$ не будет клик на $\alpha_i + 1$ вершине и степени всех вершин в $G(V_i)$ будут не больше α_i .

Отметим, что (c, p) -невырожденность раскраски является весьма сильным условием. Даже в случае двудольного графа G возникают определенные трудности с раскраской графа в Δ_G цветов. Если зафиксировать количество цветов D , в которое мы красим граф, не ограничивая при этом максимальную степень графа, то утверждение *теорема 3.1* становится неверной, причем уже для $c = 2$ и любого значения p .

Контрпример. В качестве первой доли G возьмем множество S_1 , состоящее из $(p - 1)D + 1$ элемента, в качестве второй доли S_2 возьмем множество всех p -элементных подмножеств множества S_1 и соединим каждую такую выборку со всеми входящими в нее элементами из S_1 (см. рис. 3.1). Попытаемся раскрасить получившийся двудольный граф в D цветов. Заметим, что по принципу Дирихле, в множестве S_1 при любой раскраске графа G в D цветов найдется p вершин одного цвета, а значит соответствующая выборка во второй доли графа G будет смежна с вершинами только одного цвета.

Замечание. К сожалению, величина $p(c) = (c^3 + 8c^2 + 19c + 6)(c + 1)$ для маленьких c дает достаточно большое значение. Вполне возможно, что даже с помощью нашего метода доказательства можно получить лучшую оценку. Однако, нашим методом получить асимптотику лучшую, чем $c^4(1 + O(c^{-1}))$, не получится.

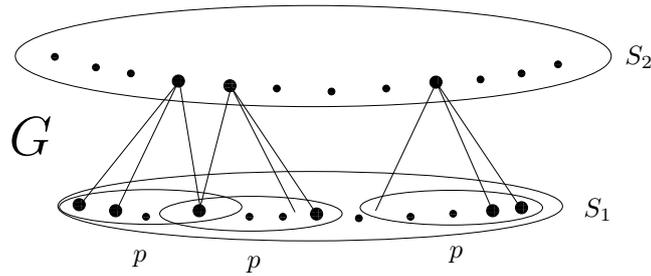


Рис. 3.1: граф G с $|S_2| = \binom{|S_1|}{p}$

Утверждение 3.2.1. Не умаляя общности в теореме 3.1 можно считать, что степени всех вершин графа G не меньше p .

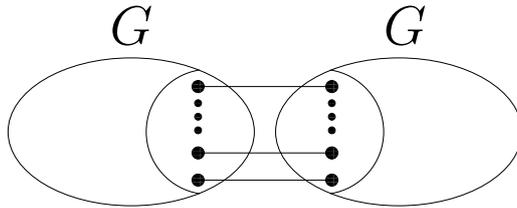


Рис. 3.2: Новый граф с большей минимальной степенью

Доказательство. С графом G можно проделать следующую операцию: взять две копии графа G и соединить в этих копиях все пары одинаковых вершин, степени которых меньше p (см. рис. 3.2).

Заметим, что, для нового графа выполнены все условия *теоремы 3.1*. Также заметим, что, если нам удастся покрасить правильным образом новый граф в D цветов с выполнением условия, что в окрестности

каждой вершины степени хотя бы p будет с различных цветов, то мы автоматически раскрасим правильным образом в D цветов обе копии графа G с выполнением все того же условия. Прделавав такую операцию несколько раз в итоге мы получим граф, в котором не будет вершин степени меньше p , ибо при каждой такой операции минимальная степень графа увеличивалась (если, конечно, в графе были вершины степени меньшей, чем p). \square

Доказательство *теоремы 3.1* состоит из двух частей. В первой части мы сведем нашу теорему, попутно доказав *теорему 3.3*, к некоторой лемме (см. лемму 3.2.1). А во второй части мы собственно докажем эту лемму.

3.2.1 Часть первая.

Покрасим граф G произвольным образом в цвета $1, 2, \dots, c + 1$. Поставим в соответствие каждому цвету $i \in \{1, 2, \dots, c + 1\}$ число α_i , где $\alpha_i = \lceil \frac{D}{c+1} \rceil$, либо $\alpha_i = \lfloor \frac{D}{c+1} \rfloor$, причем $\sum_{i=1}^{c+1} \alpha_i = D$. Понятно, что мы всегда сможем так выбрать α_i . Рассмотрим величину Φ , которая вычисляется для данной раскраски графа G в цвета $\{1, 2, \dots, c + 1\}$ следующим образом: $\Phi := \sum_{i=1}^{c+1} \frac{f_i}{\alpha_i}$, где f_i – это количество ребер графа G , соединяющих две вершины i -ого цвета. Рассмотрим такие раскраски графа G в $c + 1$ цвет, что значение Φ для них минимально. Обозначим множество всех таких раскрасок через G_c . Очевидно, что G_c не пусто. Обозначим множество всех вершин i -ого цвета любой раскраски $\xi \in G_c$ через ϑ_i . Тогда для любой раскраски ξ графа G из множества G_c верны следующие утверждения.

Утверждение 3.2.2. Для любого цвета $i \in \{1, 2, \dots, c + 1\}$ в раскраске ξ и любой вершины $v \in \mathcal{V}_i$, количество вершин цвета i смежных с v не превосходит α_i .

Доказательство. Предположим противное. Тогда, в силу условия $\sum_{j=1}^{c+1} \alpha_j = D$ найдется такой цвет $j \neq i$, что v смежна в графе G менее чем с α_j вершинами цвета j . Перекрасив v в цвет j , мы получим меньшее значение Φ . Противоречие. \square

Утверждение 3.2.3. Если какая-нибудь вершина $v \in \mathcal{V}_i$ смежна в раскраске ξ ровно с α_i вершинами цвета i , то тогда для любого цвета j вершина v смежна ровно с α_j вершинами цвета j .

Доказательство. Предположим противное. Тогда, в силу условия $\sum_{k=1}^{c+1} \alpha_k = D$, найдется такой цвет $j' \neq i$, что v смежна в графе G менее чем с $\alpha_{j'}$, а тогда, перекрасив v в цвет j' , мы уменьшим значение Φ . Противоречие. \square

Утверждение 3.2.4. Если в раскраске ξ вершина $v \in \mathcal{V}_i$ смежна хотя бы с одной вершиной своего цвета, то она смежна хотя бы с одной вершиной любого другого цвета.

Доказательство. В противном случае перекрасив v в цвет, с которым она не соединена, мы придем к противоречию с минимальностью $\Phi(\xi)$. \square

Докажем теперь, что в множестве раскрасок G_c есть раскраска, в которой ни для какого цвета i нет клик размера $\alpha_i + 1$ в графе G на вершинах i -ого цвета. Назовем такие клики **большими**.

Заметим, что одноцветных клик большего размера не может быть в силу утверждения 3.2.2.

Для любой раскраски ξ из G_c обозначим через $\phi(\xi)$ количество *больших* клик в ξ . Обозначим через Ω множество раскрасок из G_c с минимальным количеством *больших* клик. Пусть $\phi > 0$ для раскрасок из Ω .

Используя утверждение 3.2.3, получим:

Утверждение 3.2.5. *Если взять вершину v из какой-нибудь большой клики в какой-то раскраске $g_c \in \Omega$ и перекрасить ее в любой другой цвет, то, во-первых, получится новая раскраска $g'_c \in G_c$, во-вторых, $\phi(g'_c) \leq \phi(g_c)$.*

Так как ϕ мы взяли минимальным, то количество *больших* клик останется неизменным. Значит, обязательно появится новая *большая* клика цвете, в которой мы перекрасили v , кроме того получаем $g'_c \in \Omega$.

Утверждение 3.2.6. *Пусть раскраска $\xi_1 \in \Omega$ и $\phi(\xi_1) > 0$. Пусть C_1 – большая клика i -ого цвета. Рассмотрим индуцированный подграф G_{ij} графа G на всех вершинах i -ого и j -ого цвета. Тогда компонента связности в графе G_{ij} содержащая C_1 будет представлять собой полный граф размера $\alpha_i + \alpha_j + 1$.*

Доказательство. Перекрасим произвольную вершину $v_1 \in C_1$ в цвет j . Согласно утверждению 3.2.5, мы получим новую раскраску $\xi_2 \in \Omega$. При этом вершина v_1 войдет в какую-то *большую* клику C_2 j -ого цвета. Перекрасим какую-нибудь отличную от v_1 вершину v_2 клики C_2 в цвет i . Опять, согласно утверждению 3.2.5, получим новую раскраску $\xi_3 \in \Omega$, в которой v_2 обязательно должна войти в какую-то *большую* клику C_3

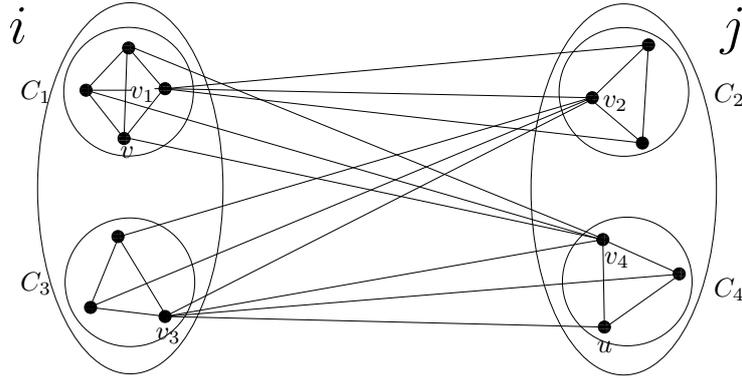


Рис. 3.3: процесс перекрашивания для $k = 4$, $\alpha_i = \alpha_j = 3$.

i -ого цвета. И так далее: мы будем перекрашивать вершины пока не получим *большую* клику, часть вершин из которой мы уже рассматривали ранее в ходе этих перекрашиваний (см. рис. 3.3, где всего было сделано четыре перекрашивания и $\alpha_i = \alpha_j = 3$). Очевидно, что если на $k + 1$ -ом шаге мы получили *большую* клику с какими-то вершинами из $V(C_l - v_l)$, где $l \leq k$, то $V(C_{k+1}) \supset V(C_l - v_l)$.

1.а) Пусть мы в конце вернулись к части клики C_1 , причем самих перекрашиваний было хотя бы три, то есть последняя раскраска была ξ_k , где $k \geq 3$ (см. рис. 3.3). Попробуем в изначальной раскраске ξ_1 перекрасить в клике C_1 какую-нибудь отличную от v_1 вершину v в цвет j . По *утверждению 3.2.5* получится *большая* клика цвета j . Легко видеть, что эта клика содержит все вершины C_k кроме v_{k-1} , следовательно, любая вершина $v \in C_1$, где $v \neq v_1$, смежна со всеми вершинами в C_k кроме вершины v_{k-1} , так как v смежна с вершиной v_k .

Отсюда можно сделать вывод, что любая вершина $u \in C_k$, где $u \neq v_{k-1}$, смежна со всеми вершинами из C_1 , кроме v_1 .

Перекрасим вершину v в раскраске ξ_1 в j -ый цвет, а затем перекрасим некоторую вершину u из C_k , отличную от v_{k-1} и v_k , в i -ый цвет (мы так можем сделать, так как из определения α_i и α_j следует, что $\alpha_i \geq 2$ и $\alpha_j \geq 2$). Получим раскраску $\xi' \in G_c$ с меньшим ϕ , так как u не смежна с v_1 и смежна со всеми остальными вершинами из C_1 . Следовательно, данный случай невозможен.

1.b) Заметим, что если мы вернулись к части клики C_1 и перекрашиваний вершин было всего два, то тогда вершина v_2 смежна со всеми вершинами из клики C_1 , а значит при перекраске любой вершины из *большой* клики C_1 в j -ый цвет мы получим по *утверждению 3.2.5* новую *большую* клику j -ого цвета, содержащую $V(C_2) \setminus \{v_1\}$. Таким образом, $G(V(C_1) \cup V(C_2))$ – полный граф. В силу произвольности выбора v_1 и v_2 и того, что граф $G(V(C_1) \cup V(C_2))$ – клика размера $\alpha_i + \alpha_j + 1$ в G_{ij} , мы получаем, что вершины из множества $V(C_1) \cup V(C_2)$ не смежны с другими вершинами i -ых и j -ых цветов.

2) Если же мы вернулись в результате перекрашиваний к клике C_l , то исходя из всего выше сказанного (мы можем считать, что мы начали процесс перекрашивания с ξ_l) $G(V(C_l) \cup V(C_{l+1}))$ образует клику в графе G . А тогда мы получаем $l = 1$, так как вершины из множества $V(C_l) \cup V(C_{l+1})$ не смежны с другими вершинами i -ого и j -ого цвета.

□

Замечание. Отметим, что в доказательстве *утверждения 3.2.6* мы существенно пользовались тем, что $\alpha_i \geq 2$ и $\alpha_j \geq 2$. Иначе мы просто не смогли бы выбрать отличную от всех v_i вершину.

Утверждение 3.2.7. *В любой раскраске $g_c \in \Omega$ нет больших клик.*

Доказательство. Пусть у этой раскраски есть *большая* клика C , например, первого цвета. Применим *утверждение 3.2.6* к первому и второму цветам. Получим полный граф на вершинах цвета 1 и 2 размера $\alpha_1 + \alpha_2 + 1$ содержащий C . Заметим, что мы можем произвольным образом разбить этот полный граф на две части цвета 1 и 2 размера $\alpha_1 + 1$ и α_2 соответственно, при этом полученная новая раскраска все равно будет лежать в Ω . Используя *утверждение 3.2.6* для первого и i -ого цвета ($i \in [2, c + 1]$) и предыдущее соображение легко доказать наличие в графе G полного подграфа на $1 + \sum_{j=1}^{c+1} \alpha_j$ вершине, то есть полного подграфа размера $D + 1$ – противоречие с условием *теоремы 3.1*. \square

Замечание. На самом деле мы только что доказали *теорему 3.3*. Также отметим, что искомая в *теореме 3.3* раскраска ξ задает разбиение всех вершин графа на нужные в *следствии* множества.

Замечание. Рассмотрим конкретную раскраску $g_c \in \Omega$. Мы только что показали, что в раскраске g_c нет *больших* клик. Таким образом, используя теорему Брукса, мы получим правильную раскраску каждого подграфа i -ого цвета в α_i цветов, в результате чего мы покрасим граф G правильным образом в D цветов ($\sum_{j=1}^{c+1} \alpha_j = D$). Если какая-то вершина в раскраске g_c была смежна с вершиной своего цвета, то по *утверждению 3.2.4* у нее в окрестности (уже в раскраске в D цветов) будет хотя бы $c + 1$ различных цветов. То есть основная проблема заключается в раскраске “вырожденных” вершин, то есть вершин, несмежных со своим и еще каким-нибудь цветом, раскраски g_c . Действительно, если бы G был изначально двудольным, теорема о его раскраске в D цветов с выполнением условия про “большое” количество различных цветов в окрестности каждой вершины была бы совсем неочевидной. На самом деле доказа-

тельность этой теоремы для случая двудольного графа весьма наглядно показывает, в чем собственно состоит сложность и специфика данной задачи, поэтому этот случай полезно иметь в виду.

Рассмотрим раскраску $g_c \in \Omega$ и рассмотрим в ней все вершины, которые смежны с вершинами не более чем $c - 1$ различных цветов. Обозначим множество всех таких вершин Υ . Заметим, что любая вершина $v \in \Upsilon$, во-первых, не смежна ни с одной вершиной своего цвета в раскраске g_c , и, во-вторых, есть еще один отличный от цвета v цвет такой, что v не смежна ни с одной вершиной этого цвета. Таким образом, эту вершину $v \in \Upsilon$ всегда можно перекрасить в другой цвет, так чтобы полученная новая раскраска по прежнему была из Ω . Более того, можно перекрасить все или только часть вершин из Υ некоторого i -ого цвета так, чтобы полученная раскраска по прежнему была из Ω (естественно, перекрашивать мы их будем скорее всего в различные цвета).

Для любой вершины $v \in \Upsilon$ найдется такой цвет в раскраске g_c , что v соединена по крайней мере с $\lceil \frac{p}{c-1} \rceil$ вершинами этого цвета. Разобьем Υ на $c + 1$ множество $\theta_1, \theta_2, \dots, \theta_{c+1}$, таким образом, что любая вершина из θ_i смежна по крайней мере с $\lceil \frac{p}{c-1} \rceil$ вершинами именно i -ого цвета.

Обозначим для всех $i \in [1, c + 1]$ через H_i индуцированный подграф графа G на множестве вершин i -ого цвета в раскраске g_c .

Утверждение 3.2.8. *Для любой вершины $v \in H_i$ верно следующее неравенство: $\left[d_{H_i}(v) + \frac{d_{G(\theta_i \cup \{v\})}(v)}{c+2} \right] \leq \alpha_i$.*

Доказательство. Рассмотрим множество E_v всех ребер в графе G с одним концом в вершине v . Очевидно, что $|E_v| \leq D$. Рассмотрим множе-

ство $E_1 \subseteq E_v$ всех ребер из E_v , у которых смежная с v вершина не лежит в θ_i .

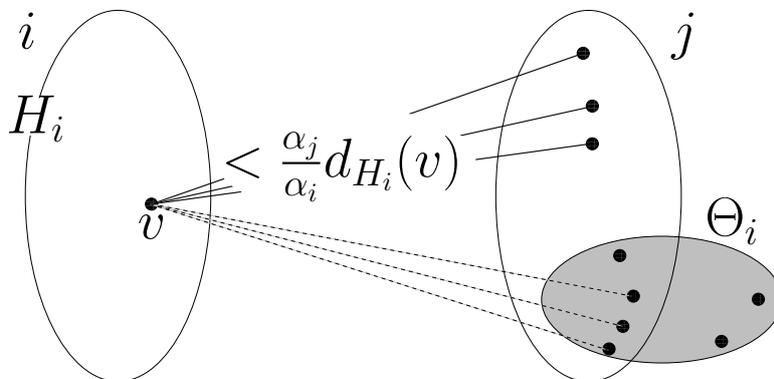


Рис. 3.4: из вершины v выходит меньше, чем $\frac{\alpha_j}{\alpha_i} d_{H_i}(v)$, ребер E_1

Пусть из v ведет в некоторый цвет j отличный от i меньше, чем $\frac{\alpha_j}{\alpha_i} d_{H_i}(v)$ ребер из множества E_1 (см. рис. 3.4). Тогда перекрасим все вершины j -ого цвета множества $\theta_i \subseteq \Upsilon$ так, чтобы полученная раскраска по-прежнему была из Ω . Очевидно, мы перекрасим эти вершины не в цвет i , поэтому величина $d_{H_i}(v)$ в новой раскраске не изменит своего значения. Перекрасим в новой раскраске v в j -ый цвет. Заметим, что к значению Φ для полученной раскраски добавится величина меньшая чем $\frac{\alpha_j}{\alpha_i} d_{H_i}(v) - \frac{d_{H_i}(v)}{\alpha_i} = 0$, таким образом мы нашли раскраску с меньшим значением Φ — противоречие.

Тогда получаем, что всего из вершины v в графе G ведет по крайней

мере R ребер, где

$$\begin{aligned} R &= d_{H_i}(v) + d_{G(\theta_i \cup \{v\})}(v) + \sum_{j \neq i} \frac{\alpha_j}{\alpha_i} d_{H_i}(v) \\ &= \sum_{j=1}^{c+1} \frac{\alpha_j}{\alpha_i} d_{H_i}(v) + d_{G(\theta_i \cup \{v\})}(v) = \frac{D}{\alpha_i} d_{H_i}(v) + d_{G(\theta_i \cup \{v\})}(v). \end{aligned}$$

По определению $R \leq |E_v| \leq D$. Получаем:

$$D \geq \frac{D}{\alpha_i} d_{H_i}(v) + d_{G(\theta_i \cup \{v\})}(v) \Rightarrow \alpha_i \geq d_{H_i}(v) + \frac{\alpha_i}{D} d_{G(\theta_i \cup \{v\})}(v).$$

Используя то, что $\alpha_i \geq \lfloor \frac{D}{c+1} \rfloor$ и тот факт, что $D \geq (c^3 + 8c^2 + 19c + 6)(c + 1)$, получим $\frac{\alpha_i}{D} > \frac{1}{c+2}$. А тогда

$$\alpha_i \geq d_{H_i}(v) + \frac{d_{G(\theta_i \cup \{v\})}(v)}{c+2}.$$

□

3.2.2 Часть вторая.

Лемма 3.2.1. Пусть дано два непустых множества вершин A и B и некоторый связный граф $H = (A \cup B, E)$, обозначим через \mathcal{G} индуцированный подграф $H(B)$. Определим $d_A(v)$, где $v \in B$, как количество ребер ведущих из вершины v в множество A . Пусть граф H обладает следующими свойствами:

- 1) никакие две вершины множества A не соединены ребром;
- 2) степень каждой вершины из A в графе H по крайней мере $q^3 + 2q^2 - q - 8$, где $q \geq 4$;
- 3) для любой вершины $v \in B$ выполняется неравенство:

$$d_{\mathcal{G}}(v) + \left\lceil \frac{d_A(v)}{q} \right\rceil \leq d, \quad (1)$$

где $d \geq q^3 + 2q^2 - q - 8$.

Тогда можно раскрасить граф \mathcal{G} правильным образом в d цветов так, чтобы для любой вершины $v \in A$ среди ее соседей в B были вершины по крайней мере q различных цветов.

Замечание. $(c + 2)^3 + 2(c + 2)^2 - (c + 2) - 8 = c^3 + 8c^2 + 19c + 6$.

В лемме 3.2.1 множество вершин B обозначает то, что в части 1 обозначалось как H_i , множество вершин A обозначает то, что раньше обозначалось как θ_i , причем нам абсолютно все равно были ли между вершинами из θ_i какие-нибудь ребра. Нам важно только с какими вершинами из H_i были смежны вершины из θ_i , ибо красить мы будем только вершины из H_i .

Через q мы обозначим в лемме 3.2.1 то, что в части 1 было равно $c + 2$, а через d обозначим то, чему раньше было равно α_i . Через H в лемме 3.2.1 мы обозначим граф $G(\theta_i \cup H_i) - E(G(\theta_i))$. По определению множества θ_i из любой вершины $v \in \theta_i$ вело по крайней мере $\frac{p}{c-1} > q^3 + 2q^2 - q - 8$ ребер в множество вершин $V(H_i)$. Мы будем полагать в лемме 3.2.1, что граф H связан (можно доказать все отдельно для каждой из компонент связности). Кроме того можно считать, что множество θ_i не пусто, иначе мы получаем обычную теорему Брукса, так как тогда нам нужно всего лишь покрасить правильным образом граф H_i в α_i цветов, а мы знаем, что в графе H_i нет полных подграфов на $\alpha_i + 1$ вершине (в графе H_i нет больших клик) и что $d_{H_i}(v) = \left\lceil d_{H_i}(v) + \frac{d_{G(\theta_i \cup \{v\})}(v)}{c+2} \right\rceil \leq \alpha_i$. Таким образом все условия леммы 3.2.1 выполнены для множеств $B = V(H_i)$ и $A = \theta_i$.

Если мы, используя лемму 3.2.1 для множеств $B = V(H_i)$ и $A = \theta_i$,

каждый подграф H_i графа G в раскраске g_c раскрасим правильным образом в α_i новых цветов, так, что каждая вершина из θ_i будет смежна с вершинами по крайней мере s различных цветов в графе H_i , то тогда мы получим правильную раскраску в D цветов всего графа G , при этом, как легко заметить, все вершины из множества $\Upsilon = \bigcup_{i=1}^{c+1} \theta_i$ будут смежны с вершинами по крайней мере s различных цветов. Кроме того в силу определения множества Υ очевидно, что все вершины из множества $V(G) \setminus \Upsilon$ будут смежны по крайней мере s различными группами цветов, а значит и просто с s вершинами различных цветов. Таким образом мы свели *теорему 3.1* к *лемме 3.2.1* и нам осталось только доказать *лемму 3.2.1*.

Замечание. Вся вторая часть будет посвящена доказательству *леммы 3.2.1*. Поэтому, дабы избежать непонимания, связанного с совпадением обозначений, сразу оговорим то, что обозначения в **части 1** и в **части 2** не имеют никакой связи, то есть эти части автономны и не следует искать определения какой-нибудь величины из **части 2** в **части 1**.

Замечание. В утверждении *леммы 3.2.1* мы можем заменить q на $q - 2$. Но для удобства дальнейших выкладок мы этого делать не будем.

Доказательство Леммы 3.2.1. Предположим, что лемма не верна. Рассмотрим минимальный по количеству вершин граф H , удовлетворяющий всем условиям из *леммы 3.2.1*, для которого не верно утверждение *леммы 3.2.1*.

Определение 3.3. Назовем *допустимым* множество $S_i \subseteq B$, если $S_i \subset N_H(v_i)$, где $v_i \in A$ и $|S_i| = q$. Множество всех возможных наборов выборок *допустимых* множеств S_i по всем $i \in \{1, 2, \dots, |A|\}$ будем обозначать как Λ .

Утверждение нашей леммы вытекает из следующего факта:

Факт 3.2.1. *Для каждой вершины v_i из A мы сможем выбрать непустое множество S_i таким образом, что если добавить к ребрам $E(\mathcal{G})$ все полные графы на множествах S_i , то вершины полученного нового графа $\tilde{\mathcal{G}}$ можно раскрасить в d цветов правильным образом.*

Замечание. Мы будем рассматривать $\tilde{\mathcal{G}}$ как граф с кратными ребрами.

Таким образом, мы получили равносильную переформулировку леммы 3.2.1. Для доказательства раскрашиваемости $\tilde{\mathcal{G}}$ в d цветов оказывается удобным проводить следующую редукцию графа.

Пусть в нашем графе $\tilde{\mathcal{G}}$ есть некоторая вершина \hat{v} степени меньше d , тогда можно “выкинуть” эту вершину из графа $\tilde{\mathcal{G}}$ и доказывать все уже для графа $\tilde{\mathcal{G}} \setminus \hat{v}$. Если мы сможем покрасить редуцированный граф, то потом мы всегда сможем докрасить \hat{v} по принципу Дирихле в свободный цвет.

Определение 3.4. Будем говорить, что \hat{v} рекурсивно удаляется из графа $\tilde{\mathcal{G}}$, если можно провести последовательность таких редукций, чтобы последней в этой последовательности редуцировалась вершина \hat{v} . А граф $\tilde{\mathcal{G}}$ будем называть рекурсивным, если он последовательно редуцируется до пустого графа.

На самом деле мы будем доказывать более сильный факт, а именно: вместо утверждения про то, что $\tilde{\mathcal{G}}$ красится правильным образом в d цветов, мы будем доказывать, что $\tilde{\mathcal{G}}$ является рекурсивным относительно раскраски в d цветов.

Вернемся к доказательству нашей леммы, а точнее к доказательству последней переформулировки факта 3.2.1. Обозначим через S множе-

ство тех вершин из B , которые смежны хотя бы с одной вершиной из A .

Докажем, что для графа H будет выполнена усиленная переформулировка факта 3.2.1, в предположении, что H минимальный по количеству вершин граф, для которого не верна лемма 3.2.1. Таким образом мы получим противоречие, а значит докажем лемму 3.2.1.

Определение 3.5. Определим для каждой вершины v из множества B величину

$$L(v) := d_g(v) + \frac{d_A(v)}{q+1}.$$

Факт 3.2.2. Заметим, что если выбрать S_i случайным образом (независимо для каждой вершины v_i , причем для каждой вершины v_i все возможные варианты множества S_i равновероятны), то математическое ожидание степени каждой вершины v из множества S в графе $\tilde{\mathcal{G}}$ будет не больше $d_g(v) + d_A(v)(q-1)\frac{q}{q^3+2q^2-q-8}$, то есть не более $L(v)$ (так как $q \geq 4$, то $q^3 + 2q^2 - q - 8 > q(q^2 - 1)$), а значит по третьему условию леммы 3.2.1 меньше d . Таким образом, в среднем степень каждой вершины в $\tilde{\mathcal{G}}$ будет меньше d . А это дает нам надежду на то, что граф $\tilde{\mathcal{G}}$ окажется рекурсивным, то есть, если мы последовательно будем удалять из $\tilde{\mathcal{G}}$ вершины степени меньше чем d , мы в конце концов получим пустой граф.

Для завершения доказательства леммы нам осталось удачно выбрать S_i (т.е. выбрать так, чтобы $\tilde{\mathcal{G}}$ стал рекурсивным).

Определение 3.6. Сдвигом нескольких допустимых множеств $S_{i_1}, S_{i_2}, \dots, S_{i_z}$ в выборке $\lambda \in \Lambda$ назовем замену λ на $\lambda' \in \Lambda$, где λ' отличается от λ только тем, что все допустимые множества S_{i_1}, \dots, S_{i_z} заменены в ней

на некоторые другие *допустимые* множества $S'_{i_1}, \dots, S'_{i_z}$. Множества $S'_{i_1}, \dots, S'_{i_z}$ будем называть *результатом сдвига* множеств $S_{i_1}, S_{i_2}, \dots, S_{i_z}$.

Обозначим через R множество $B \setminus S$. Можно считать, что степень любой вершины из R равна d , так как по условию *леммы 3.2.1* степень любой вершины из B в графе \mathcal{G} меньше либо равна d , а если степень вершины меньше d , то можно сразу выкинуть рекурсивно эту вершины из $\tilde{\mathcal{G}}$, для любой допустимой выборки.

Утверждение 3.2.9. Пусть дан граф F такой, что $V(F) = F_1 \cup F_2$ и $F_1 \cap F_2 = \emptyset$. Степень любой вершины из F_2 в графе F меньше либо равна D и в F существует вершина $v \in F_1$, что

- граф $F(F_2 \cup \{v\})$ *связен*,
- $d_F(v) < D$,
- v смежна со всеми остальными вершинами в F_1 .

Пусть кроме того, граф $F(F_1)$ уже раскрашен правильным образом в D цветов. Тогда имеющуюся раскраску можно дополнить до правильной раскраски графа F в D цветов.

Доказательство. Выкинем из графа F (см. рис. 3.5) вершину v , получится новый граф F' . На множестве $F_1 \setminus \{v\}$ уже есть правильная раскраска в D цветов. По одной вершине рекурсивно докрасим правильным образом все вершины из F_2 , так можно сделать в силу связности графа $F(F_2 \cup \{v\})$ и условия, что степень в графе F любой вершины из F_2 меньше либо равна D . Перенесем полученную правильную раскраску графа F' в D цветов на F и покрасим v в какой-нибудь цвет отличный от всех цветов вершин из $N_F(v)$ (так можно сделать, так как по

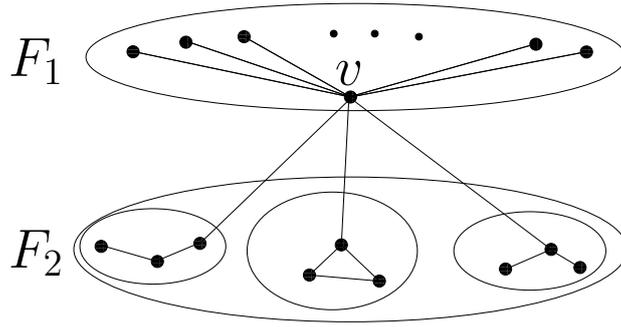


Рис. 3.5: граф F

условию $d_F(v) < D$). В итоге мы покрасили правильным образом в D цветов граф F , правда при этом мы возможно изменили цвет у вершины v по сравнению с заданной раскраской на F_1 . Заметим, что все вершины из множества $F_1 \setminus \{v\}$ будут покрашены в цвета отличные от цвета v в исходной раскраске F_1 , так как эта раскраска была правильной для графа $F(F_1)$, а вершина v смежна со всеми остальными вершинами в F_1 , кроме того заметим, что все цвета вершин из множества $F_1 \setminus \{v\}$ отличны от цвета v в раскраске F . Теперь, если мы изменили цвет у вершины v по сравнению с исходной раскраской F_1 , то просто поменяем цвет, совпадающий с цветом v в исходной раскраске F_1 , и нынешний цвет v в раскраске F местами. Получится опять правильная раскраска F , но уже полностью совпадающая с исходной раскраской на множестве F_1 .

□

Определение 3.7. *Регулярным сдвигом* множеств S_i , некоторого набора $\nu \in \Lambda$ относительно множества S' мы будем называть такой *сдвиг* множеств S_i , где $i \in [1, |A|]$, в множества S'_i , $i \in [1, |A|]$, что для всех

$i \in [1, |A|]$ множество $S_i \cap S'$ содержит множество $S'_i \cap S'$. Если найдется $i \in [1, |A|]$, что $|S_i \cap S'|$ строго больше $|S'_i \cap S'|$, то такой регулярный сдвиг мы будем называть *невырожденным*.

Замечание. Регулярный сдвиг относительно некоторого множества есть регулярный сдвиг относительно любого подмножества этого множества, правда невырожденность при этом не обязательно сохраняется.

Утверждение 3.2.10. Пусть есть набор допустимых множеств $\eta = \{S_1, S_2, \dots, S_{|A|}\}$ графа H — минимального по количеству вершин контр-примера к лемме 3.2.1, и пусть есть такие множества $S' \subseteq S, R' \subseteq R$, что все вершины из $B \setminus (S' \cup R')$ рекурсивно удаляются из графа $\tilde{\mathcal{G}}$, причем для всех $u \in R$ $d_{\tilde{\mathcal{G}}}(u) = d$ и для всех $u \in R'$ $d_{\tilde{\mathcal{G}}(S' \cup R')}(u) = d$.

Кроме того пусть $\tilde{H}' := \tilde{\mathcal{G}}(S' \cup R')$ и

$$\sum_{u \in \tilde{H}'} d_{\tilde{H}'}(u) > \sum_{u \in \tilde{H}'} L(u). \quad (2)$$

Тогда можно регулярно сдвинуть невырожденным образом несколько множеств S_i относительно множества $S' \cup R'$, так чтобы $B \setminus (S' \cup R')$ по-прежнему рекурсивно удалялось уже из полученного в результате этого сдвига графа $\tilde{\mathcal{G}}'$.

Доказательство. Доказывать это утверждение мы будем индукцией по размеру множества $B \setminus (S' \cup R')$.

База случай: $|B \setminus (S' \cup R')| = 0$ очевидно невозможен, так как в силу факта 3.2.2 не будет выполнено условие (2).

Переход от всех меньших, чем k , размеров множества $B \setminus (S' \cup R')$ к размеру k .

Пусть $Z := B \setminus (S' \cup R')$ (см. рис. 3.6).

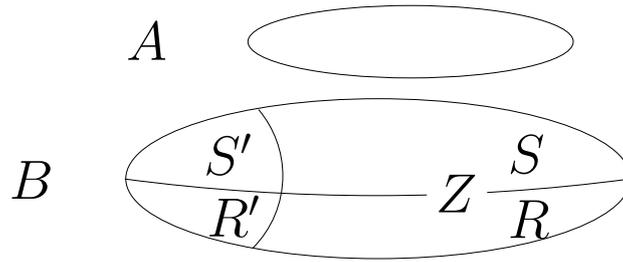


Рис. 3.6: граф H

Пусть у нас есть множества S' и R' такие, что $|Z| = k$ и для них не выполнено утверждение индукционного перехода.

Покажем, что существует такая вершина $v_i \in A$ и соответствующее ей множество S_i , что S_i можно регулярно сдвинуть относительно S' невырожденным образом. Предположим противное, тогда для любого допустимого множества S_i и соответствующей ему вершины $v_j \in A$ допускается всего две возможности:

- 1) множество $S_i \cap S' = \emptyset$ (см. рис. 3.7);
- 2) множество $N_H(v_i) \setminus S' \subseteq S_i$ (см. рис. 3.8).

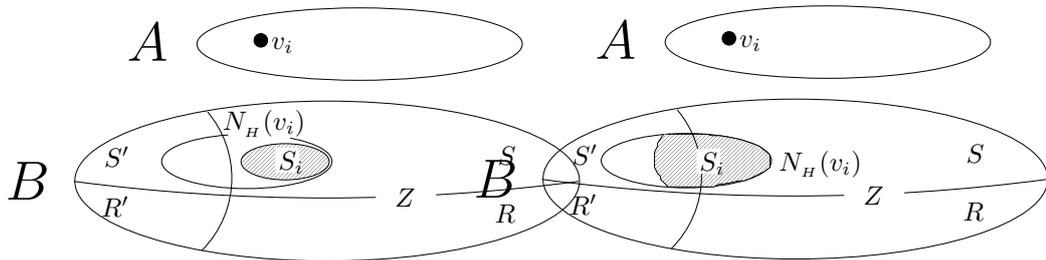


Рис. 3.7: $S_i \cap S' = \emptyset$

Рис. 3.8: $N_H(v_i) \setminus S' \subseteq S_i$

Как нетрудно заметить, количество добавленных ребер с концами в

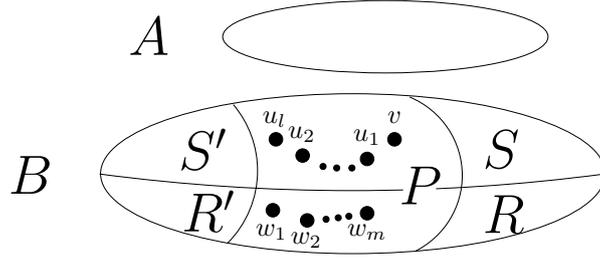


Рис. 3.9: $P = \{v, u_1, u_2, \dots, u_l, w_1, w_2, \dots, w_m\}$

множестве вершин S' в обоих этих случаях достигает своего минимального значения. Таким образом верна цепочка неравенств $\sum_{v \in V(\tilde{H}')} d_{\tilde{H}'}(v) \leq \sum_{v \in V(\tilde{H}')} E(d_{\tilde{G}}(v)) \leq \sum_{v \in V(\tilde{H}')} L(v)$, где $E(\cdot)$ обозначает математическое ожидание случайной величины (с оговоренным в факте 3.2.2 распределением). Мы знаем из условия (2), что $\sum_{u \in \tilde{H}'} d_{\tilde{H}'}(u) > \sum_{u \in \tilde{H}'} L(u)$. Получаем противоречие с предыдущим неравенством.

Значит все-таки есть такая вершина $v_i \in A$, что часть ее окрестности находящаяся в Z не лежит в соответствующем множестве S_i , а множество $S_i \cap S' \neq \emptyset$.

Таким образом мы можем рассмотреть такую вершину $v \in N_H(v_i)$, что она не лежит ни в множестве S' , ни в множестве S_i , при этом часть множества S_i лежит в множестве S' . Мы знаем, что Z рекурсивно удаляется из $\tilde{\mathcal{G}}$. Начнем удалять *рекурсивно* вершины из Z , причем будем *рекурсивно* удалять пока это возможно вершины отличные от v . В какой-то момент мы не сможем ничего удалить, кроме v . Это значит, что от множества Z остались вершины $v, u_1, u_2, \dots, u_l \in S, w_1, w_2, \dots, w_m \in R$ (см. рис. 3.9).

Обозначим множество всех оставшихся в Z вершин через P , а через \tilde{I} обозначим индуцированный граф $\tilde{\mathcal{G}}(S' \cup R' \cup P)$.

Заметим, что степени в графе \tilde{I} всех вершин u_k , где $k \in [1, l]$, и всех вершин w_j , где $j \in [1, m]$, по крайней мере d .

Заметим также, что степень вершины v в \tilde{I} меньше d .

Если степень вершины v меньше $d - q + 1$, то сдвинем S_i в множество S'_i следующим образом: возьмем вершину x из S_i , которая лежит в множестве $S' \cap S_i$ (такая вершина x обязательно найдется, т.к. $S' \cap S_i \neq \emptyset$), тогда $S'_i := (S_i \setminus \{x\}) \cup \{v\}$. Остальные множества из набора η мы двигать не будем. Заметим, что описанный нами сдвиг будет регулярным и невырожденным относительно множества S' , также понятно, что множество Z по-прежнему можно рекурсивно удалить (очевидно, что можно также как и раньше рекурсивно удалить все вершины из $Z \setminus P$, потом можно рекурсивно удалить вершину v , т.к. степень вершины v в графе \tilde{I} была меньше $d - q + 1$, а значит после сдвига стала не более $d - 1$, а потом можно рекурсивно удалить оставшиеся в Z вершины, т.к. множество Z рекурсивно удалялось из графа $\tilde{\mathcal{G}}$ и новых ребер в графе $\tilde{I} - \{v\}$ при описанном выше сдвиге не добавилось). В этом случае мы доказали индукционный переход.

Предположим противное, пусть степень v меньше d , но по крайней мере $d - q + 1$.

Докажем, что уже для графа \tilde{I} выполнено неравенство

$$\sum_{u \in \tilde{I}} d_{\tilde{I}}(u) > \sum_{u \in \tilde{I}} L(u). \quad (2')$$

Пусть мы доказали это, тогда мы можем применить предположение индукции для множеств S'_0 и R'_0 , где $S'_0 := (S' \cup R' \cup P) \cap S$ и $R'_0 :=$

$(S' \cup R' \cup P) \cap R$, то есть мы можем регулярно сдвинуть невырожденным образом набор множеств η относительно $S' \cup R' \cup P$, так чтобы вершины из $B \setminus (S' \cup R' \cup P)$ по-прежнему удалялись рекурсивно в полученном графе. Заметим, что если набор *сдвинулся регулярно* относительно $S' \cup R' \cup P$, то он *сдвинулся регулярно* относительно $S' \cup R'$. Кроме того заметим, что в полученном в итоге сдвига графе будут по-прежнему рекурсивно удаляться все вершины из множества $B \setminus (S' \cup R')$, так как мы можем вначале рекурсивно удалить все вершины множества $B \setminus (S' \cup R' \cup P)$, а затем удалить рекурсивно все вершины из P , благо что новых ребер в графе \tilde{I} в результате такого сдвига не добавилось.

Мы будем так делать пока одно из множеств S_i регулярно не сдвинется невырожденным образом относительно $S' \cup R'$, либо пока степень вершины v не станет меньше $d - q + 1$, либо степень какой-то вершины из $P \setminus \{v\}$ не станет меньше d . В последнем случае мы рекурсивно сможем еще выкинуть несколько вершин из Z и уже для нового меньшего графа \tilde{I} применить те же самые рассуждения. Здесь необходимо отметить, что рано или поздно мы обязательно придем к одному из этих вариантов, так как иначе мы будем бесконечное число раз делать регулярный невырожденный сдвиг относительно множества $S' \cup R' \cup P$, а значит сколь угодно долго уменьшать значение суммы $\sum_{i=1}^{|A|} |S_i \cap (S' \cup R' \cup P)|$.

Обозначим через l' количество ребер, ведущих из множества вершин P в множество вершин S' .

В силу условий *утверждения 3.2.10*, что для всех $u \in R'$ $d_{\tilde{g}(S' \cup R')}(u) = d$, и что для всех $u \in R$ $d_{\tilde{g}}(u) = d$, не может быть ребер между множествами P и R' .

Для завершения доказательства *утверждения 3.2.10* осталось дока-

зять, что для графа $\tilde{I} = \tilde{\mathcal{G}}(S' \cup R' \cup P)$ выполнено неравенство (2'). Предположим противное. Тогда

$$\sum_{u \in \tilde{I}} L(u) \geq \sum_{u \in \tilde{I}} d_{\tilde{I}}(u) \geq \sum_{u \in \tilde{H}'} d_{\tilde{H}'}(u) + l' + \sum_{u \in P} d_{\tilde{I}}(u) > \sum_{u \in \tilde{H}'} L(u) + \sum_{u \in P} d_{\tilde{I}}(u) + l'.$$

То есть мы получаем, что

$$\sum_{u \in P} L(u) > l' + \sum_{u \in P} d_{\tilde{I}}(u).$$

А значит мы получаем неравенство:

$$l' + \sum_{u \in P} (d_{\tilde{I}}(u) - L(u)) < 0. \quad (3)$$

Оценим величину $d_{\tilde{I}}(u_i) - L(u_i)$ для всех $i \in [1, l]$.

По определению $L(u)$ и в силу того, что $d_{\tilde{I}}(u_i) \geq d$ получаем, что для любого $i \in [1, l]$ верно неравенство $d_{\tilde{I}}(u_i) - L(u_i) \geq d - d_{\mathcal{G}}(u_i) - \frac{d_A(u_i)}{q+1}$. Используя неравенство (1) получим:

$$d_{\tilde{I}}(u_i) - L(u_i) \geq d - d_{\mathcal{G}}(u_i) - \frac{d_A(u_i)}{q+1} \geq d_{\mathcal{G}}(u_i) + \left\lceil \frac{d_A(u_i)}{q} \right\rceil - d_{\mathcal{G}}(u_i) - \frac{d_A(u_i)}{q+1}. \quad (*)$$

Таким образом $d_{\tilde{I}}(u_i) - L(u_i) \geq \left\lceil \frac{d_A(u_i)}{q} \right\rceil - \frac{d_A(u_i)}{q+1}$. Далее $d_A(u_i) > 0$ для всех $i \in [1, l]$, так как $u_i \in S$ для всех $i \in [1, l]$. Рассмотрим два случая:

- а) $0 < d_A(u_i) \leq q$;
- б) $d_A(u_i) \geq q + 1$.

В обоих случаях выполнено неравенство:

$$d_{\tilde{\Gamma}}(u_i) - L(u_i) \geq \lceil \frac{d_A(u_i)}{q} \rceil - \frac{d_A(u_i)}{q+1} \geq \frac{1}{q+1}. \quad (4)$$

Пусть $q_1 := d_{\tilde{\Gamma}}(v) - d + q$, тогда, как мы показали выше, $q_1 > 0$. Заметим, что для вершины v аналогично выкладкам (*) и (4) можно получить неравенство

$$d_{\tilde{\Gamma}}(v) - L(v) \geq q_1 - q + \frac{1}{q+1}. \quad (5)$$

Так как $w_i \in R$, где $i \in [1, m]$, то тогда $d_{\mathcal{G}}(w_i) = L(w_i)$, кроме того вершины из множества $P \cap R$ нельзя *рекурсивно* удалить из графа $\tilde{\Gamma}$. Используя еще условие *утверждения 3.2.10*, что для любой вершины $u \in R$ $d_{\mathcal{G}}(u) = d$, мы получаем $d_{\tilde{\Gamma}}(w_j) = d_{\mathcal{G}}(w_j) = d$. А тогда для всех $i \in [1, m]$ верно равенство:

$$d = d_{\tilde{\Gamma}}(w_i) = L(w_i). \quad (6)$$

Подставим теперь неравенства (4), (5) и равенство (6) в неравенство (3). Получим неравенство:

$$l' + l \frac{1}{q+1} - q + q_1 + \frac{1}{q+1} < 0.$$

Напомним, что l это количество вершин в множестве $(P \cap S) \setminus \{v\}$, то есть количество u_i . Мы знаем, что $q_1 \geq 1$. Тогда $l'(q+1) + l < (q-1)(q+1) - 1$, то есть

$$(q+1)l' + l \leq q^2 - 3. \quad (7)$$

Очевидным образом из неравенства (7) получаем неравенство

$$l \leq q^2 - 3 \quad (8)$$

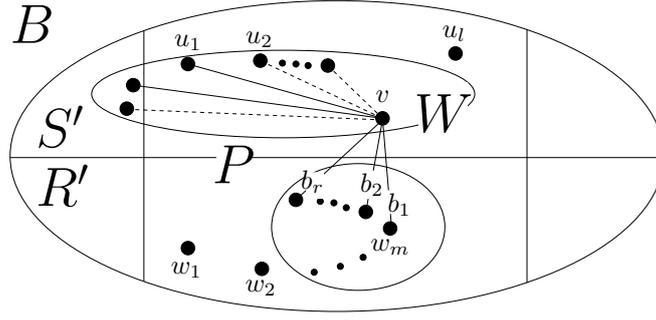


Рис. 3.10: множество вершин W смежных с чем-то из C_v

и неравенство

$$l' \leq q - 2 . \quad (9)$$

Обозначим через b_j , где $j \in [1, r]$, все вершины из множества $R \cap V(\tilde{I})$, в которые ведет ребро из вершины v (r может быть равно 0). Разберем несколько случаев.

1) $r \geq q^2 - 3$.

Обозначим через C_v объединение всех компонент связности графа $\tilde{\mathcal{G}}(R)$, содержащих все вершины b_j , где $j \in [1, r]$. Как отмечалось ранее, между множествами P и R' не может быть ребер, значит $C_v \subseteq R \setminus R'$ (см. рис. 3.10). По равенству (6) мы имеем $d_{\tilde{I}}(w_j) = d$, где $j \in [1, m]$. Таким образом вершины из множества $P \cap R$ не смежны с вершинами $Z \setminus P$, а значит $C_v \subseteq \{w_1, w_2, \dots, w_m\}$.

Рассмотрим в множестве вершин S графа $\tilde{\mathcal{G}}$ все смежные с какой-нибудь вершиной из C_v . Обозначим множество всех таких вершин через W (см. рис. 3.10). Очевидно в силу равенств $d_{\tilde{I}}(w_i) = d$ и $d_{\mathcal{G}}(w_i) = d$, что, во-первых, $W \subseteq V(\tilde{I})$, во-вторых, $v \in W$ и, в-третьих, для всех вершин

$u \in C_v$, верно равенство $d_{\tilde{g}(W \cup C_v)}(u) = d$.

$$1.1) |W| \geq q^2 - 1.$$

Тогда $|W \cap S'| = |W| - |W \cap P \cap S| \geq q^2 - 1 - (l + 1)$. Очевидно, что $l' \geq |W \cap S'|$. Таким образом $l' \geq q^2 - 2 - l$, то есть $l' + l \geq q^2 - 2$. Получаем противоречие с неравенством (7).

$$\text{Таким образом мы показали, что } |W| \leq q^2 - 2.$$

$$1.2) |W| \leq q^2 - 2.$$

Проведем в графе H дополнительно все ребра вида (u, v) (см. рис. 3.10), где $u \in W$ и $(u, v) \notin E(H)$. Получим новый граф Θ . Проверим теперь, что все условия *леммы 3.2.1* выполнены для графа $\hat{H} := \Theta(V(\Theta) \setminus C_v)$ с множествами $\hat{A} := A$, $\hat{B} := B \setminus C_v$ и величиной $\hat{d} := d$.

Условие 1) очевидно, в силу того, что множество A осталось прежним.

Условие 2) очевидно, в силу того, что вершины из множества C_v не были смежны ни с одной вершиной из множества A .

Условие 3). Неравенство (1) достаточно проверить только для вершин из множества B , для которых мы проводили новые ребра с концами в этих вершинах, то есть проверять нужно для всех вершин из множества W . По определению $\hat{\mathcal{G}} := \hat{H}(\hat{B})$. Для любой вершины $u \in W$, $u \neq v$ мы добавили к графу \mathcal{G} максимум одно ребро инцидентное u , но с другой стороны мы убрали по крайней мере одно ребро, ведущее в множество C_v (W по определению — множество всех вершин из S , смежных с какой-нибудь вершиной из C_v). Таким образом для таких вершин $u \in W$, что $u \neq v$ неравенство (1) осталось верным. Для вершины v неравенство (1) будет выполнено потому, что $|W| \leq q^2 - 2$, значит мы

добавили не более, чем $q^2 - 3$ ребра. С другой стороны, множество C_v по определению содержит все вершины b_j , где $j \in [1, r]$, $r \geq q^2 - 3$, а значит мы выкинули по крайней мере $q^2 - 3$ ребра с концом в v .

Так как мы предполагали, что H минимальный по количеству вершин граф, для которого не выполнена *лемма 3.2.1*, то *лемма 3.2.1* выполнена для графа \hat{H} , в котором меньше вершин чем в H . Тогда можно правильным образом покрасить граф $\hat{\mathcal{G}}$ в d цветов так, чтобы для любой вершины $u \in A$ среди ее соседей в \hat{B} было по крайней мере q различных цветов. Обозначим такую раскраску ξ . Легко видеть, что все предположения из формулировки *утверждения 3.2.9* выполнены для графа $\Phi := \Theta(W \cup C_v)$, множеств $F_1 := W$ и $F_2 := C_v$, $D := d$ и вершины v . Рассмотрим правильную раскраску $\xi(W)$ графа $\Theta(W)$ в d цветов. По *утверждению 3.2.9* раскраску $\xi(W)$ можно продолжить до правильной раскраски ζ в d цветов графа $\Theta(W \cup C_v)$. Заметим, что в графе Θ нет ребер между множеством вершин C_v и множеством вершин $V(H) \setminus (W \cup C_v)$, поэтому раскраски ξ и ζ можно объединить в одну правильную раскраску графа \mathcal{G} в d цветов, кроме того, очевидно, что для любой вершины $u \in A$ в этой раскраске среди ее соседей в B будет по крайней мере q различных цветов. Таким образом мы получили нужную в *лемме 3.2.1* раскраску для графа H – противоречие с условием *утверждения 3.2.10*.

2.) $r \leq q^2 - 4$

Таким образом в графе \tilde{I} из вершины v ведет не более, чем $q^2 - 4$ ребра в множество вершин $R \cap V(\tilde{I})$. Степень вершины v в графе \tilde{I} равна $d - q + q_1$. Значит из вершины v ведет как минимум $d - q + q_1 - r - l'$ ребер в множество вершин $\{u_1, u_2, \dots, u_l\}$.

Заметим, что если у какой-то вершины $u \in S$ в графе $\tilde{\mathcal{G}}$ есть ребро кратности k , то величина $d_A(u) \geq k - 1$. Мы знаем, что в вершины u_1, u_2, \dots, u_l из v идет по крайней мере $q^3 + 2q^2 - q - 8 - q + q_1 - r - l'$ ребер. Обозначим через d_i кратность ребра (v, u_i) в графе \tilde{I} для всех $i \in [1, l]$. Мы знаем, что $\sum_{i=1}^l d_i \geq q^3 + 2q^2 - 2q - 8 + q_1 - r - l'$. Тогда получаем, что

$$\sum_{i=1}^l d_A(u_i) \geq \sum_{i=1}^l (d_i - 1) \geq q^3 + 2q^2 - 2q - 8 + q_1 - r - l' - l. \quad (10)$$

Подставив неравенство (5) и равенство (6) в неравенство (3), получим:

$$l' - q + q_1 + \frac{1}{q+1} + \sum_{i=1}^l (d_{\tilde{I}}(u_i) - L(u_i)) < 0. \quad (11)$$

Мы знаем, что $d_{\tilde{I}}(u_i) - L(u_i) \geq d - d_{\mathcal{G}}(u_i) - \frac{d_A(u_i)}{q+1}$, для всех $i \in [1, l]$. Применив неравенство (1), получаем $d - d_{\mathcal{G}}(u_i) - \frac{d_A(u_i)}{q+1} \geq \lceil \frac{d_A(u_i)}{q} \rceil - \frac{d_A(u_i)}{q+1} \geq d_A(u_i) \frac{1}{q(q+1)}$. Таким образом

$$d_{\tilde{I}}(u_i) - L(u_i) \geq d_A(u_i) \frac{1}{q(q+1)}. \quad (12)$$

Подставив (12) в (11), получим:

$$l' - q + q_1 + \frac{1}{q+1} + \frac{1}{q(q+1)} \sum_{i=1}^l d_A(u_i) < 0. \quad (13)$$

Подставим (10) в (13):

$$l' - q + q_1 + \frac{1}{q+1} + \frac{q^3 + 2q^2 - 2q - 8 + q_1 - r - l' - l}{q(q+1)} < 0.$$

Мы знаем, что $l' \geq 0$ и $q_1 \geq 1$. Таким образом, имеем:

$$l'(1 - \frac{1}{q(q+1)}) - (q-1) + \frac{1}{q+1} + \frac{q^3 + 2q^2 - 2q - 8 + 1 - r - l}{q(q+1)} < 0 .$$

Мы знаем, что $r \leq q^2 - 4$, и по неравенству (8), $l \leq q^2 - 3$. Тогда

$$\frac{1}{q+1} + \frac{q^3 + 2q^2 - 2q - 8 + 1 - (q^2 - 4) - (q^2 - 3) - (q^3 - q)}{q(q+1)} < 0 .$$

То есть мы получили, что $\frac{1}{q+1} - \frac{q}{q(q+1)} < 0$ – противоречие.

Таким образом мы доказали неравенство (2') для графа \tilde{I} , а значит *утверждение 3.2.10*. □

Вернемся к доказательству *леммы 3.2.1*. Для конкретного набора множеств $\lambda \in \Lambda$ начнем рекурсивно удалять, пока это возможно, вершины из графа $\tilde{\mathcal{G}}$. Так как граф $\tilde{\mathcal{G}}$ не рекурсивный, то в конце от графа $\tilde{\mathcal{G}}$ останется граф $\tilde{\mathcal{G}}(S' \cup R')$, где $S' \subseteq S$, $R' \subseteq R$. Выберем среди всех наборов из Λ такой набор λ , что величина $|S'| + |R'|$ для него принимает минимальное значение. Заметим, что для множеств S' и R' выполнены все предположения *утверждения 3.2.10*. Единственное неочевидное место в проверке этих условий – проверка неравенства (2).

Так как из графа $\tilde{H}' := \tilde{\mathcal{G}}(S' \cup R')$ нельзя больше ничего рекурсивно удалить, то степени всех вершин в нем по крайней мере d . По неравенству (1) $L(u) = d_{\mathcal{G}}(u) + \frac{d_A(u)}{q+1} \leq d$ для всех вершин $u \in B$, причем $L(u) < d$ для всех вершин u из множества S . А так как множество S' очевидно не пусто, то

$$\sum_{u \in \tilde{H}'} d_{\tilde{H}'}(u) \geq d|\tilde{H}'| > \sum_{u \in \tilde{H}'} L(u).$$

Будем применять *утверждение 3.2.10* к множествам S' и R' , до тех пор пока не получим такой вершины, что ее степень в \widetilde{H}' меньше d (заметим, что мы не можем делать бесконечное число *регулярных невырожденных* сдвигов относительно множества $S' \cup R'$, так как при каждом таком сдвиге величина $\sum_{i=1}^{|A|} |S_i \cap S'|$ уменьшается). Заметим, что в силу *утверждения 3.2.10* мы по-прежнему сможем рекурсивно удалить все вершины из B , кроме вершин $S' \cup R'$, а тогда мы сможем рекурсивно удалить еще и эту вершину степени меньше d . Получаем противоречие с тем, что мы выбрали набор с минимальной величиной $|S'| + |R'|$. Значит, найдется такой допустимый набор, что граф $\widetilde{\mathcal{G}}$ для этого набора будет рекурсивным. Таким образом мы доказали *лемму 3.2.1* и в конце концов *теорему 3.1*. □

Глава 4

Алгоритмы для Раскрасок Графов

Основная цель настоящей работы заключается в описании полиномиального алгоритма для d -списковых раскрасок. Мы также приводим здесь доказательство результата Бородина, как следствие нашего алгоритмического результата.

Алгоритмические аспекты. Хотя для общей задачи списковой раскраски трудно найти даже приближенное решение, известен линейный по времени алгоритм предложенный Скулраттанакулчайя [25] для любого набора списков размера Δ , который обобщает алгоритмический результат Ловаша [23] аналогичной сложности. Алгоритм Скулраттанакулчайя появился как следствие нового доказательства теоремы Ердеша, Рубина и Тейлора [16]. В отличие от последнего, мы в нашем алгоритме опираемся только на простую идею рекурсивного удаления вершин, представленную Дираком и Ловашем [10] в альтернативном доказательстве

теоремы Брукса [14]. Эта идея позволила нам представить первый линейный по времени алгоритм для задачи d -списковой раскрасок. Из-за того, что d -списковая раскраска подчиняется простым правилам редукции, наш алгоритм может быть применен с тем же успехом к задаче дополнения частичной Δ -списковой раскраски. Таким образом настоящая работа обобщает в нескольких направлениях результат [25], где приводится первый линейный по времени алгоритм для Δ -списковых раскрасок. Как часть алгоритма мы описываем линейную по времени процедуру для обычной Δ -раскраски адаптированную к применению внутри основного алгоритма, которая может быть применена к общей задаче Δ раскраски графа, ко всему прочему, эта процедура порождает алгоритмическое доказательство теоремы Брукса и кажется довольно полезной на практике.

Реализацию всего алгоритма, вместе с процедурой, можно найти в [29]. В заключение хотелось бы отметить, что наша работа может быть рассмотрена как алгоритмическое и весьма нетривиальное дополнение к результату Бородина [11].

4.1 Обозначения

Таким образом нам даны граф и списковый набор \mathcal{L} и наша задача заключается в том, чтобы либо найти правильную \mathcal{L} -раскраску, либо показать ее отсутствие. В проблеме расширения \mathcal{L} -раскраски нам дано, в дополнение к G и \mathcal{L} , множество $S \subset V$ и правильная раскраска $G(S)$, тогда наша задача заключается в том, чтобы расширить эту \mathcal{L} -раскраску на все вершины графа G или показать, что этого сделать нельзя.

Граф называется *d-выбираемым*, если для любого спискового *d*-набора \mathcal{L} задача \mathcal{L} -раскраски имеет решение.

4.1.1 Терминология

Мы будем предполагать, что граф G связан, так как задача о раскраске несвязного графа может быть решена отдельно для каждой компоненты связности.

Напомним некоторые известные факты касающиеся блоков и точек сочленения, которые можно найти например в [10] или в [15]:

- Два блока могут иметь не более одной общей вершины и если они пересекаются, то эта вершина обязательно будет точкой сочленения G .
- Любое ребро G входит ровно в один блок.
- Двудольный граф, где первая доля соответствует блокам G , вторая точкам сочленения G , и точки сочленения смежны только с содержащими их блоками, является деревом.

Такое дерево называется *деревом блоков и точек сочленения*.

Предположим у нас есть остовное дерево T графа G с корнем r , кроме того ребра в дереве имеют ориентацию в направлении корня. Такие деревья называются *нормальными*, если найдется направленный путь в T соединяющий любые две вершины $u, v \in V$ если $(u, v) \in E$. В [15] доказано, что любой связный граф содержит нормальное дерево. Легко видеть, что дерево поиска в глубину является нормальным. Используя этот факт, легко показать, что корень нормального дерева двусвязного

графа является висячей вершиной, так как иначе при ее удалении мы получим более, чем одну компоненту связности.

4.1.2 Основной результат

Перед тем как обсуждать алгоритм, мы опишем несколько естественных требований на входные данные. Мы рассматриваем наиболее сжатое задание графа, где вершины представлены целыми числами от 1 до $N = |V|$, а ребра $E(G)$ заданы как множество пар.

Определение 4.1. Пусть максимальное значение цветов для \mathcal{L} не больше чем $c|E(G)|$, для некоторой константы c . Тогда мы говорим, что \mathcal{L} удовлетворяет *условию плотности цветов*.

Последнее условие нужно нам, чтобы эффективно работать с цветами. Условие это довольно слабое, так как позволяет описать всевозможные различные ситуации для \mathcal{L} с точностью до перенумерации цветов. Такое же условие на количество цветов и нумерацию вершин также накладывалось в [25].

Теорема 4.1. *Существует линейный по времени алгоритм, который для данных графа G и функции спискового d -набора \mathcal{L} , удовлетворяющей условию плотности цветов, либо находит \mathcal{L} -раскраску G , либо сообщает о ее несуществовании.*

Наш алгоритм эффективным образом основывается на следующей теореме, описывающей все наборы списков цветов и графов, для которых существует допустимая раскраска.

Теорема 4.2. Пусть $G = (V, E)$ граф, с множеством блоков \mathcal{B} , и набором списков цветов \mathcal{L} . Тогда G не имеет списковой \mathcal{L} -раскраски тогда и только тогда, когда:

1. Каждый $B \in \mathcal{B}$ является либо кликой, либо нечетным циклом.
2. Существует отображение \mathcal{S} ставящее в соответствие каждой паре (v, B) , где $v \in B \in \mathcal{B}$, множество цветов $\mathcal{S}(v, B)$ удовлетворяющее следующим условиям:

$$(a) \quad \bigcup_{B: v \in B \in \mathcal{B}} \mathcal{S}(v, B) = \mathcal{L}(v)$$

$$(b) \quad \mathcal{S}(v, B) \cap \mathcal{S}(v, B') = \emptyset \text{ для любых } B \neq B' \in \mathcal{B}, \text{ где } v \in B \cap B'.$$

$$(c) \quad |\mathcal{S}(v, B)| = \deg_{G(B)}(v) \text{ для любой } v \in B$$

$$(d) \quad \mathcal{S}(v, B) = \mathcal{S}(u, B) \text{ для любых } u, v \in B$$

Замечание. Условия теоремы 2a и 2c с необходимостью влекут 2b; условие 1 есть ни что иное, как характеристика Ердеша d -выбираемых графов. Отметим, что появление клик и нечетных циклов здесь не случайно и связано с теоремой Брукса [14], говорящей, что для связного графа H не являющегося ни кликой, ни нечетным циклом выполнено $\chi(H) \leq \Delta(H)$.

Замечание. Эта теорема является переформулировкой результата Бородина [11]. Еще один вариант результата Бородина можно найти в [21].

Другими словами, теорема 4.2 утверждает, что не раскрашиваемые примеры задачи списковой d -раскраски можно описать следующим образом. В деревьях блоков и точек сочленения G мы можем разделить список каждой точки сочленения v на t непересекающихся частей и оставить для каждого блока содержащего v ровно одну часть таким образом, что

после этого ни один блок нельзя покрасить правильным образом используя только цвета из оставленной части списка соответствующую данному блоку (для вершин – не точек сочленения, мы отсавляем тот же список). (См. рис. 4.1)

Эту характеристику полезно иметь в виду для понимания самого алгоритма и последующего анализа. Алгоритм для теоремы 4.1 описывается в секции 4.2.

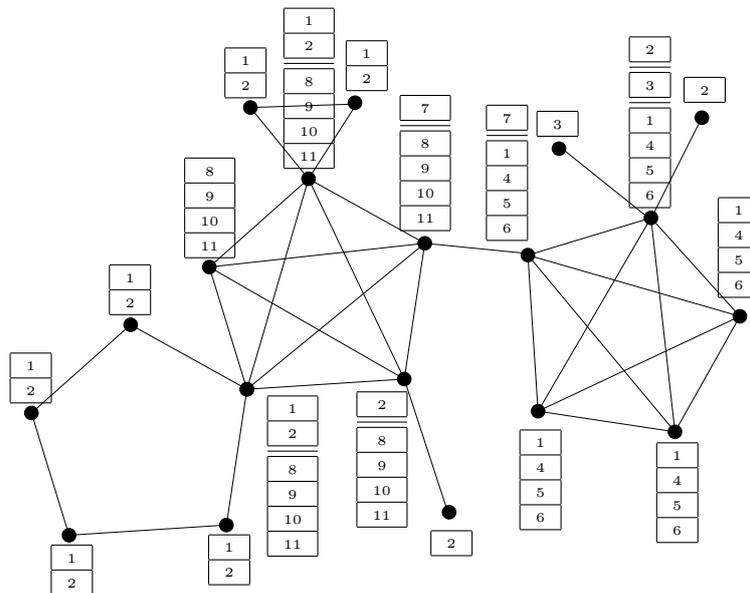


Рис. 4.1: Пример графа и набора \mathcal{L} списков, у которого нет правильной \mathcal{L} -раскраски. Список цветов каждой вершины разделен на некоторое число меньших списков, согласно теореме 4.2.

4.2 Задача Списковой d -раскраски

ВХОД: Граф $G = (V, E)$ и набор списков \mathcal{L} , где $\mathcal{L}(v) \geq deg_G(v)$ для любой вершины $v \in V$.

ЗАДАЧА: Найти правильную \mathcal{L} -раскраску G или показать, что такой не существует.

Чтобы упростить дальнейшее изложение и последующий анализ алгоритма, мы будем считать, что G связан. Однако, почти такой же алгоритм работает для несвязных графов.

Наш алгоритм существенно использует простое наблюдение, что если в G есть вершина v для которой $|\mathcal{L}(v)| > \deg_G(v)$, то можно очень легко покрасить весь граф. В алгоритме мы используем некоторое обобщение этой идеи: мы пытаемся найти вершину u и некоторый ее допустимый цвет c , что, если u покрасить в этот цвет c и удалить его из списков цветов всех соседей u , то оставшийся граф $G - u$ будет связным и кроме того в нем найдется вершина v удовлетворяющая $|\mathcal{L}(v)| > \deg_{G-u}(v)$.

Общая схема алгоритма выглядит грубо говоря следующим образом. Сначала в G мы находим дерево блоков и точек сочленения T . На следующем шаге мы берем блок B соответствующий висячей вершине T . Используя как вспомогательный инструмент наблюдение описанное выше, мы либо красим весь граф, либо находим раскраску B , за исключением соответствующей точки сочленения v_0 (в этом случае B должен быть или кликой, или нечетным циклом). В последнем случае мы удаляем $B \setminus \{v_0\}$ из G , также как и все цвета соседей v_0 из $\mathcal{L}(v_0)$. Мы повторяем этот шаг до тех пор, пока либо не раскрасим граф, либо не обнаружим вершину с пустым списком цветов.

4.2.1 Алгоритм

Дадим несколько определений прежде чем начать описание алгоритма.

Определение 4.2. Назовем блок *висячим*, если он содержит не более одной точки сочленения, т.е. соответствует висячей вершине в дереве блоков и точек сочленения \mathcal{T} .

Определение 4.3. Назовем упорядоченную пару смежных вершин (u, v) *различимыми цветом i* , если $i \in \mathcal{L}(u) \setminus \mathcal{L}(v)$.

В основном алгоритме мы воспользуемся следующими вспомогательными процедурами:

1. **Рекурсивное удаление вершин.** Находит искомую d -раскраску связного графа G , если G имеет вершину v с $|\mathcal{L}(v)| > \deg_G(v)$.
2. **Нахождение блоков в G .** Находит представление каждого блока в виде множества его вершин.
3. **Поиск цвета различающего пару (u, v) .** Требуется от нас ввести один бинарный массив \mathcal{C} размера $s|E(G)|$, который мы будем использовать многократно при каждом применении процедуры.
4. **Δ -раскраска связного графа.** Для связного графа H , который не является ни кликой, ни нечетным циклом, процедура находит правильную раскраску в Δ цветов.

Теперь мы наконец готовы дать описание нашего алгоритма.

- **ШАГ 1.** Пускай найдется вершина $v \in V$ с $|\mathcal{L}(v)| > \deg_G(v)$. Процедура **рекурсивного удаления вершин** выдаст нам искомую раскраску и в этом случае мы закончим выполнение алгоритма.
- **ШАГ 2. А)** Находим все блоки в G и строим дерево блоков и точек сочленения \mathcal{T} .

1. Находим блоки G используя процедуру **поиска блоков**.
 2. Для каждой вершины v графа находим список блоков, содержащих v .
 3. Строим двудольный граф H_0 , где одна часть состоит из блоков, а другая часть состоит из всех точек сочленения с множеством ребер $E(H_0) = \{(v, B) \mid v \in B\}$.
 4. Строим дерево блоков и точек сочленения \mathcal{T} в порядке обхода *поиска в ширину*, т.е. применяя поиск в ширину находим соответствующий порядок \mathcal{B} на блоках и точках сочленения, в зависимости от момента начала их обработки в поиске.
- **В)** Берем последний блок B в порядке \mathcal{B} и единственную точку сочленения в этом блоке v_0 . В следующих шагах мы пытаемся уменьшить G удалив вершины $B \setminus \{v_0\}$, или найти желаемую раскраску.
 - **ШАГ 3.** Пытаемся найти в B пару различных некоторым цветом вершин (u, v) таких, что u не является точкой сочленения, т.е. $u \neq v_0$.
 1. Если в B имеется такая пара вершин (u, v) различных цветом c , тогда мы красим u в цвет c и удаляем ее из G . В оставшемся графе $|\mathcal{L}(v)| > \deg(v)$, кроме того $G' = G \setminus \{u\}$ связан. Находим раскраску G' с помощью процедуры **рекурсивного удаления вершин** и завершаем работу.
 2. Иначе мы переходим к следующему шагу.
 - **ШАГ 4.** Проверяем является ли блок B кликой или нечетным циклом. Это можно легко проверить предварительно посчитав степени всех вершин в G .

1. Если B является кликой или нечетным циклом, мы уменьшаем G удаляя $B \setminus \{v_0\}$. Из за того, что в ШАГЕ 3 v_0 и любая вершина v блока B смежная с v_0 не различимы никаким цветом, мы имеем $\mathcal{L}(v_0) \supseteq \mathcal{L}(v)$. Удалим из списка $\mathcal{L}(v_0)$ список $\mathcal{L}(v)$. Затем вернемся к ШАГУ 2 В) и применим его к идущему до \mathcal{B} в порядке B блоку. В этом месте алгоритма мы пользуемся свойством порядка \mathcal{B} , чтобы эффективно уменьшить $\mathcal{L}(v_0)$.
2. Другие варианты для B . В этом случае мы обязательно получим желаемую раскраску и завершим работу.
 - Раскрасим $G(V \setminus V(B))$ применив процедуру **рекурсивного удаления вершин** к $G(V \setminus B)$.
 - Удалим из $\mathcal{L}(v_0)$ все цвета, в которые покрашены соседи v_0 .
 - Проверим одинаковые ли списки допустимых цветов у v_0 и любой смежной с ней вершиной u из B .
 - Если эти списки разные, тогда пара (v_0, u) для оставшихся не покрашенных вершин будет различима некоторым цветом. Дополним раскраску всего графа аналогично ШАГУ 3.1 и завершим работу.
 - В противном случае применим процедуру **Δ -раскраски** к B .

4.2.2 Реализация процедур

Здесь мы подробно описываем наши вспомогательные процедуры и проводим анализ сложности их работы.

Рекурсивное удаление вершин

Пусть есть $v \in V$ с $|\mathcal{L}(v)| > \deg_G(v)$. Поместим вершину v в стек \mathcal{O} и удалим ее из графа.

В оставшемся графе опять найдется вершина удовлетворяющая тому же самому свойству, так как G связан и $|\mathcal{L}(u)| \geq \deg(u)$ для каждой вершины $u \in V$. Повторим предыдущее действие для новой вершины, т.е. поместим ее в \mathcal{O} и удалим из G . Продолжим удалять вершины до тех пор, пока в G не останется вершин.

Возьмем первую вершину из \mathcal{O} и покрасим ее в произвольный допустимый цвет, затем возьмем вторую вершину и опять покрасим ее в любой допустимый цвет, который еще не встречался среди ее соседей и так далее (в результате мы сможем покрасить любую вершину в виду подходящего выбора \mathcal{O}). В результате мы покрасим все вершины G .

Замечание. Процедуру можно с тем же успехом применять и для графов имеющих более одной компоненты связности, если в каждой компоненте есть вершина w , для которой выполнено неравенство $|\mathcal{L}(w)| > \deg(w)$.

Утверждение 4.2.1. *Рекурсивное удаление вершин в G и последующая раскраска работают за линейное от размера графа время.*

Доказательство. Мы можем вычислить степени всех вершин графа G за линейное время. Действительно, после того как мы поместили вершину в \mathcal{O} , в \mathcal{O} можно поместить любого из соседей данной вершины. Таким образом мы можем запустить поиск в глубину начиная с вершины, у которой степень меньше, чем размер списка, и помещать в \mathcal{O} соответствующую вершину на каждом шаге нашего поиска. Все эти операции работают линейное время.

Чтобы покрасить все вершины в \mathcal{O} за линейное время, достаточно покрасить данную вершину $u \in \mathcal{O}$ за время $O(|\mathcal{L}(u)|)$.

Так как у нас имеется оценка сверху на максимальный номер цвета, то мы можем в стадии инициализации алгоритма за линейное время создать бинарный массив \mathcal{C}_0 заполненный изначально 0, размер которого равен максимальному номеру цвета встречающемуся среди списков цветов.

На стадии раскраски вершины u в нашей процедуре мы помечаем 1 в \mathcal{C}_0 те цвета, в которые уже покрасили соседей u , и затем ищем в $\mathcal{L}(u)$ цвет c со значением 0 в $\mathcal{C}_0[c]$. После того как мы нашли c , мы обнуляем \mathcal{C}_0 , проходя один раз по всем соседям u . Таким образом, мы возвращаем \mathcal{C}_0 в изначальное состояние и можем использовать его для других вершин. После этого мы красим u в цвет c . Завершая доказательство отметим, что для раскраски u мы использовали времени $O(|\mathcal{L}(u)|)$; также всего мы использовали $c|E(G)|$ памяти, так как создали массив такого размера всего один раз. \square

Поиск блоков в G .

Алгоритм нахождения всех блоков в графе за линейное время можно найти, например, в [26].

Поиск цвета различающего пару (u, v) .

Нам понадобится еще один массив \mathcal{C} инициализированный 0, аналогичный массиву \mathcal{C}_0 . В нем мы пометим 1 все цвета встречающиеся в $\mathcal{L}(v)$ и затем проверим есть ли цвет c в $\mathcal{L}(u)$ для которого $\mathcal{C}[c]$ равно 0; в конце процедуры заменим все 1 назад на 0. В итоге мы либо находим

различающий цвет, либо убеждаемся, что такого цвета нет, за линейное от $|\mathcal{L}(u)| + |\mathcal{L}(v)|$ время.

Нахождение Δ -раскраски связного графа.

Мы будем использовать линейный по времени алгоритм Ловаша [23] находящий Δ -раскраску за линейное время. Существует более простой альтернативный алгоритм, который мы описываем в Секции 4.3 и из которого также легко получается конструктивное доказательство теоремы Брукса.

4.2.3 Доказательство корректности алгоритма

ШАГ 1. Мы вычисляем степени всех вершин в G и для каждой вершины проверяем $|\mathcal{L}(v)| > \deg_G(v)$.

ШАГ 2.

А) Нахождение всех блоков в G и построение дерева блоков и точек сочленения \mathcal{T} . В нашем представлении блоков каждый блок описывается как множество вершин; также для каждой вершины мы строим список блоков ее содержащих. Напомним определение дерева блоков и точек сочленения. Рассматривается двудольный граф H , у которого одна доля состоит из всех возможных блоков, а дугая из всех точек сочленения (то есть вершин входящих по крайней мере в два блока) и $E(H) = \{(B, v) \mid v \in B\}$; по построению H является деревом и называется деревом блоков и точек сочленения. Из-за соображений сложности вычислений нам понадобится построить \mathcal{T} как **дерево обхода в ширину**, т.е. мы требуем, чтобы \mathcal{T} имело определенный порядок на блоках и точках сочленения.

Замечание. Мы обязаны находить все блоки в графе, так как примеры, для которых нет решения, описываются в терминах определенных условий на эти блоки [16].

Для представления дерева удобно рассмотреть ориентацию по направлению к корню. Для этого мы добавляем искусственную вершину в G и соединяем ее с произвольной вершиной. Таким образом мы добавляем искусственный блок B_0 в \mathcal{T} . Возьмем B_0 и рассмотрим порядок \mathcal{B} , в котором появляются блоки при поиске в ширину на дереве \mathcal{T} с корнем в B_0 .

В) Пусть B последний блок дерева \mathcal{T} в порядке \mathcal{B} .

Отметим, что B является висячим блоком. Обозначим через v_0 точку сочленения B , т.е. единственную точку сочленения в \mathcal{T} соединенную с B . Так как B стоит в конце \mathcal{B} и порядок \mathcal{B} соответствует обходу в ширину, все оставшиеся соседи v_0 , появляющиеся после v_0 в \mathcal{B} , являются висячими блоками.

ШАГ 3. Случай А. Предположим мы нашли упорядоченную пару вершин (u, v) различных цветом c , такую что u не является точкой сочленения в блоке B . Тогда мы красим u в цвет c и удаляем ее из G . В оставшемся графе, $|\mathcal{L}(v)| > deg(v)$. Кроме того $G' = G \setminus \{u\}$ связан, так что мы можем покрасить G' с помощью процедуры рекурсивного удаления вершин.

Случай В. Если такой пары нет, то для всех вершин не точек сочленения $w, v \in B$ имеем $\mathcal{L}(v) = \mathcal{L}(w)$, в то время как для точки сочленения v_0 имеем $\mathcal{L}(v_0) \supseteq \mathcal{L}(v)$. Следовательно, все вершины $B \setminus \{v_0\}$ имеют одинаковую степень, тогда мы переходим к ШАГУ 4.

На следующем шаге алгоритма мы либо уменьшим G отрезав $B \setminus$

$\{v_0\}$, либо найдем раскрасим G с помощью процедуры рекурсивного удаления вершин.

ШАГ 4. Списки цветов всех вершин не точек сочленения B одинаковы и содержатся в списке цветов точки сочленения B .

Согласно теореме Брукса [14] (см. Замечание 4.1.2) существует два случая, которые можно различить с помощью предварительного подсчета степеней вершин.

Случай I (B является кликой или нечетным циклом) Производить редукцию списка $\mathcal{L}(v_0)$ нужно очень осторожно из за того, что $|\mathcal{L}(v_0)|$ может быть гораздо больше, чем размер B . Мы опишем этот момент подробно в Секции 4.2.4.

Случай II (Другие варианты для B .) В этом случае согласно классификации Ердеша и других [16], мы знаем, что G обязан иметь правильную раскраску для любого набора списков.

4.2.4 Анализ сложности алгоритма

Ясно, что ШАГИ 1 и 2 занимают всего $O(|G| + |\mathcal{L}|)$ времени.

ШАГ 3.

Часть 1. Поиск пары различных некоторым цветом вершин в $B \setminus \{v_0\}$.

Утверждение 4.2.2. *Поиск пары различных некоторым цветом вершин (u, v) , таких что $u, v \neq v_0$, может быть сделан за линейное от размера блока время, где под размером блока мы подразумеваем число ребер плюс число вершин в данном блоке.*

Доказательство. Мы можем рассмотреть остовное дерево поиска в глубину блока B с корнем в v_0 и проверить только пары списков допустимых цветов для смежных вершин в этом дереве. Так как B является блоком, v_0 имеет единственного соседа в таком дереве. Таким образом для каждой пары смежных вершин u и v мы дважды применяем процедуру поиска различающего цвета, сначала для (u, v) потом для (v, u) .

Теперь покажем, что общее время всех проверок линейно по отношению к $\sum_{v \in B \setminus \{v_0\}} |\mathcal{L}(v)|$. Проверяя вершины u и v , которые не являются точками сочленения, мы либо получаем, что $\mathcal{L}(u) = \mathcal{L}(v)$, либо находим упорядоченную пару вершин и различающий их цвет. В последнем случае мы заканчиваем поиск пар. В первом же случае легко видеть, что проверка работает за линейное от $\min(|\mathcal{L}(u)|, |\mathcal{L}(v)|)$ время. Таким образом общее время работы будет линейным от $\sum_{v \in B \setminus \{v_0\}} |\mathcal{L}(v)|$. \square

Часть 2. Проверка различимы ли (u_0, v_0) некоторым цветом за линейное от $\mathcal{L}(u_0)$ время, где u_0 смежна с v_0 в B .

Замечание. Нам нужно сделать только одну проверку для v_0 и ее единственного соседа в дереве потому, что B является блоком и мы рассматриваем дерево поиска в ширину с корнем в v_0 . Но мы не можем позволить себе сделать такую проверку за линейное от $|\mathcal{L}(v_0)| + |\mathcal{L}(u_0)|$ время непосредственным образом для всех блоков висящих на v_0 . Из за того, что $|\mathcal{L}(v_0)|$ может быть намного больше, чем размер B , и более того, v_0 может быть точкой сочленения для многих маленьких блоков включая B , общее время работы может иметь квадратную асимптотику по отношению к размеру входа, например для $G = K_{n-1,1}$. По тем же причинам мы не можем непосредственным образом редуцировать $\mathcal{L}(v_0)$

в Шаге 4.

Чтобы избежать многократных проверок описанных в Замечании 4.2.4 и редукций $\mathcal{L}(v_0)$ для большого $\mathcal{L}(v_0)$ мы можем запомнить 1 значения $\mathcal{L}(v_0)$ в \mathcal{C} и использовать их не только для одного висящего на v_0 блоке, но сразу для всех таких блоков. Строго гоовря мы делаем следующее:

- Вводим с самого начала работы алгоритма новый бинарный массив \mathcal{C}' с $|\mathcal{C}'| = |\mathcal{C}|$, изначально заполненный 0.
- Если точка сочленения висячего блока B отличается от предыдущей, скажем v'_0 , тогда мы делаем следующее.
 1. Возвращаем \mathcal{C}' в изначальное состояние с одними 0 за линейное от $|\mathcal{L}(v'_0)|$ время, проходя один раз по списку $\mathcal{L}(v'_0)$.
 2. Помечаем 1 в \mathcal{C}' цвета $\mathcal{L}(v_0)$.
- Для висячего блока B и точки сочленения v_0 проверяем пару (u_0, v_0) за линейное от $|\mathcal{L}(u_0)|$ время. Это можно сделать лишь за один проход по $\mathcal{L}(u_0)$ проверяя стоит ли 1 в $\mathcal{C}'[c]$ для $c \in \mathcal{L}(u_0)$.

Чтобы найти различающий цвет для пары (u_0, v_0) , можно вместо $\mathcal{L}(v_0)$ использовать \mathcal{C}' , так как поиске различающей пары мы проверяем только принадлежит ли цвет $i \in \mathcal{L}(u_0)$ списку $\mathcal{L}(v_0)$.

ШАГ 4.

Эффективная редукция $\mathcal{L}(v_0)$. При любой редукции мы будем эффективно поддерживать только \mathcal{C}' вместо $\mathcal{L}(v_0)$, так как нам не нужен

целый список $\mathcal{L}(v_0)$, если v_0 является точкой сочленения. Таким образом при каждой редукции $\mathcal{L}(v_0)$ мы просто помечаем назад 0 все цвета $\mathcal{L}(u_0)$ в \mathcal{C}' и пока не меняем $\mathcal{L}(v_0)$. Реально мы изменим $\mathcal{L}(v_0)$ только когда обрежем все висячие блоки содержащие v_0 и когда v_0 перестанет быть точкой сочленения. В силу определения \mathcal{B} мы сделаем такую замену только один раз для v_0 и следовательно общее время потраченное на редукцию списков для всех точек сочленения займет линейное время. После описанной выше “воображаемой” замены $\mathcal{L}(v_0)$ мы раскрасим все вершины в $B \setminus \{v_0\}$ и удалим их из G . Затем вернемся назад к Шагу 2 В) и найдем следующий висячий блок в редуцированном графе.

Следовательно, в Шаге 3 и Шаге 4 алгоритм работает за время $O(|B|)$ для каждого отдельного блока B и поддержка \mathcal{C}' для каждой точки сочленения в сумме будет работать за время $O(|\mathcal{L}|)$. Складывая написанное выше, мы видим, что алгоритм работает за линейное от $O(|G| + |\mathcal{L}|)$ время.

4.3 Конструктивное доказательство теоремы Брукса

Δ -раскраска двусвязного графа, который не является ни кликой ни нечетным циклом. Здесь мы опишем алгоритм отличный от Шага 4 III. Мы можем считать, что все вершины в $B \setminus \{v_0\}$ (v_0 является единственной точкой сочленения в B) имеют один и тот же список допустимых цветов. Следовательно все вершины в $B \setminus v_0$ имеют одинаковую степень.

К этому моменту мы уже знаем, что B является блоком, т.е. двусвяз-

ным графом. Более того, мы уже нашли остовное дерево в B с порядком обхода поиска в глубину (это дерево является частью поиска в глубину для графа G) и описали процедуру *рекурсивного удаления вершин*. Используя все это мы теперь можем описать довольно простой алгоритм для раскраски G . Единственная трудность алгоритма заключается в числе комбинаторных случаев, которые нам придется рассмотреть. Таким образом описывать алгоритм мы будем в порядке рассмотрения и разбора случаев следующим образом.

- Пусть у нас имеется две несмежные вершины, скажем u и v , в $B \setminus \{v_0\}$ такие, что если мы сделаем следующее:
 - покрасим u и v в один и тот же цвет
 - удалим этот цвет из $\mathcal{L}(w)$ для каждого соседа w вершины v или u
 - удалим обе вершины u и v из G ,

то к оставшемуся графу будет применима процедура *рекурсивного удаления вершин*. Назовем такую пару вершин u и v *хорошей парой*. Следовательно, если мы нашли хорошую пару, то мы легко сможем покрасить G за линейное время.

- Если $\deg_G(u) = 2$ для некоторой вершины $u \in B \setminus \{v_0\}$, тогда B является циклом и мы знаем, что B не может быть нечетным циклом. Следовательно, B является четным циклом. Раскрасим $B \setminus \{v_0\}$ и удалим из $\mathcal{L}(v_0)$ цвет двух соседей v_0 в B и затем раскрасим оставшийся граф используя процедуру *рекурсивного удаления вершин*.

- Если в нашем дереве поиска в глубину, т.е. нормальном дереве с корнем в v_0 , есть вершина u имеющая двух потомков, то эти два потомка будут *хорошей парой*. Действительно, легко показать в силу того, что дерево с корнем в v_0 нормальное, что эти два потомка не смежны. Также не трудно убедиться, что удалив из B этих двух потомков, в оставшемся графе всегда найдется путь от u до любой другой вершины. Из этих двух соображений вытекает хорошест пары.
- Если наше дерево поиска в глубину не является простым путем, тогда мы можем эффективно, т.е. за линейное от размера блока время, найти такую пару. В последнем случае, мы завершаем работу.
- Таким образом наше дерево представляет из себя простой путь $w_0, w_1, w_2, \dots, w_{k-1}, v_0$, где v_0 , как упоминалось выше, является единственной точкой сочленения блока B в графе G . Для удобства обозначений, пусть $v_0 = w_k$. Рассмотрим ближайшую к w_0 вершину, скажем w_t , на этом пути, которая не смежна с w_0 , и следовательно $\deg_G(w_0) \geq t - 1$ (отметим, что такая вершина w_t обязательно найдется, так как блок B не является кликой и степени всех вершин в нем равны).

1. Пара (w_0, w_t) является *хорошей*, когда, либо среди w_1, w_2, \dots, w_{t-1} есть вершина, смежная с чем-то отличным от w_0 , w_1, \dots, w_t , либо $t + 1 \leq k$ и w_{t+1} смежна с w_0 .

Последнее выполняется в виду того, что w_0 и w_t не смежны и в графе $B \setminus \{w_0, w_t\}$ не может быть более двух компонент связности, так как w_0 является висячей вершиной в нормальном дереве

(w_{t-1} была смежна сразу с w_0 и w_t и вершины w_1, w_2, \dots, w_{t-1} будут связны в $B \setminus \{w_0, w_t\}$, также как v_0 будет связана со всеми остальными вершинами).

2. Пусть $\deg_G(w_0) \geq t + 1$, тогда w_1 смежна с некоторой вершиной w_l , где $l \geq t + 1$, так как $\deg_G(w_1) = \deg_G(w_0) \geq t + 1$, и, следовательно, мы нашли хорошую пару согласно случаю 1.
3. Если $t - 1 = \deg_G(w_0)$, тогда w_{t-1} не смежна с некоторой вершиной w_r , где $r < t - 1$, потому, что $\deg_G(w_0) = \deg_G(w_{t-1})$ и w_{t-1} смежна с w_t . Значит w_r и w_{t-1} образуют *хорошую пару*, так как w_r и w_{t-1} обе смежны с w_0 и w_t .
4. Предположим, что $t = \deg_G(w_0)$ и мы не в условиях случая 1. Значит w_0, w_1, \dots, w_{t-1} и w_1, \dots, w_t образуют клики, так как $t = \deg_G(w_0) = \dots = \deg_G(w_t)$ и ни одна вершина w_i , где $1 \leq i \leq t - 1$ не смежна с w_j , где $j \geq t + 1$. Также $k \geq t + 2$, так как w_0 не смежна с w_t и w_{t+1} и есть вершина w_i смежная с w_0 , где $i > t$. Мы можем считать, что $i > t + 1$, так как вариант, когда $i = t + 1$, мы рассмотрели в случае 1. Значит, так как $\deg_G(w_t) = \deg_G(w_0) > 2$, w_{t-1} и w_{t+1} образуют *хорошую пару*.

Замечание. Тот же самый алгоритм работает для Δ -раскраски двусвязного графа H . Действительно, можно взять произвольную вершину H в качестве точки сочленения v_0 , после чего раскрасить H за линейное от размера H время, в случае, если есть вершина u с $\deg_H(u) < \Delta(H)$.

Замечание. Из этого алгоритма также вытекает доказательство теоремы Брукса [14] для двусвязных графов, так как если граф не клика и не нечетный цикл, можно применить наш алгоритм и получить правильную

раскраску.

Благодарности. Мы благодарим Олега Бородина и Александра Косточку за указание весьма полезных ссылок на литературу.

Литература

- [1] J. A. Bondy, U. S. R. Murty, *Graph Theory with Applications*, American Elsevier, New York, 1976.
- [2] R. L. Brooks, *On coloring the nodes of network*, Proc. Cambridge Philos. Soc. 37 (1941) 194-197.
- [3] L. Hong-Jian, B. Montgomery, Hoifung Poon, *Upper Bounds of Dynamic Chromatic Number*, Ars. Combinatoria 68(2003), pp. 193-201.
- [4] B. Montgomery, *Dynamic Coloring*, Ph.D. Dissertation, West Virginia University, 2001.
- [5] X. Meng, L. Miao, Z. R. Li, B. Su, *The Conditional coloring numbers of Pseudo-Halin graphs*, Ars Combinatoria 79(2006), pp 3-9.
- [6] Suohai Fan, Hong-Jian Lai, Jianliang Lin, Bruce Montgomery, Zhishui Tao, *Conditional colorings of graphs*, Discrete Mathematic 16(2006),1997-2004.
- [7] H. Hind, M. Molloy, B. Reed, *Colouring a graph frugally*, Combinatorica 17(4) (1997) 469-482.
- [8] Godsil C., Royle G. *Algebraic graph theory* (Springer, 2001)(L)(T)(ISBN 0387952411).

- [9] L. Lovasz, *On decomposition of graphs*, Studia Sci. Math. Hungar. 1 (1966) 237-238.
- [10] J. A. Bondy, U. S. R. Murty, *Graph Theory with Applications*, Elsevier, New York, 1976.
- [11] O. V. Borodin, *A chromatic criteria for degree assignment, IV USSR conference on the theoretical cybernetics problems*, in Novosibirsk, proceeding reports, 127-128 (1977) (in Russian)
- [12] O. V. Borodin, *Problems of colouring and of covering the vertex set of a graph by induced subgraphs*, Ph.D. Thesis, Novosibirsk State University, Novosibirsk, 1979 (in Russian).
- [13] O. V. Borodin, A. V. Kostochka, *On an upper bound on a graph's chromatic number, depending on the graphs's degree and density* J. Comb. Th. (B) 23 (1977), p. 247-250.
- [14] R. L. Brooks, *On colouring the nodes of network*, Proc. Cambridge Phil. Soc. 37 (1941) 194–197.
- [15] R. Diestel *Graph Theory*, Springer, Berlin Heidelberg New York, 2000.
- [16] P. Erdős, A. L. Rubin, H. Taylor, *Choosability in graphs*, in: Proc. West-Coast Conf. on Combinatorics, Graph Theory and Computing, Arcata, California, Congr. Numer. XXVI (1979) 125–157.
- [17] T. Gallai, *Kritische Graphen I*, Publ. Math. Inst. Hung. Acad. Sci. 8 (1963) p. 373-395.
- [18] T. R. Jensen, B. Toft, *Graph Colouring Problems*, John Wiley & Sons, New York, 1995.

- [19] Kawarabayashi, Ken-ichi, B. Mohar, *List-color-critical graphs on a fixed surface*, SODA '09: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 1156–1165, New York 2009.
- [20] A. Kostochka, M. Stiebitz, *A list version of Dirac's Theorem*, J. Graph Th. 39 (2002).
- [21] A. Kostochka, M. Stiebitz, B. Wirth, *The colour theorems of Brooks and Gallai extended*, Discrete Math 162 299-303 (1996)
- [22] J. Kratochvíl, Z. Tuza, M. Voigt, *New trends in the theory of graph colorings: choosability and list coloring*, 183–197, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 49, Amer. Math. Soc., Providence, RI, 1999.
- [23] L. Lovász, *Three short proofs in graph theory*, J. Comb. Th.(B) 19(1975), 269–271.
- [24] B. Reed, *A strengthening of Brooks' theorem*, J. Comb. Th. (B) V. 76, (1999) p. 136 - 149.
- [25] S. Skulrattanakulchai, *Δ -List Vertex Coloring in Linear Time and Space*, Information Processing Letters, Vol. 98 , Issue 3 (May 2006), 101–106.
- [26] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. 1 (2) (1972) 146–160.
- [27] Z. Tuza, *Graph colorings with local constraints—a survey*, Discussiones Math. Graph Th. 17(2) (1997), 161–228

[28] В. Визинг, *Colouring the vertices of a graph in prescribed colours.*,
Metody Diskret. Anal. v Teorii Kodov i Schem 29 (1976), 3–10 (In
Russian)

[29] <http://www1.spms.ntu.edu.sg/~ngravin/code/Dcols.tgz>