3-раскрашиваемость за $\mathcal{O}\left(1.3478^n\right)$

immediate

12 января 2022 г.

Аннотация

Как известно 3-расрашиваемость является **NP**-полной задачей и неизвестно ее решение за полином. Мы реализуем алгоритм проверяющий граф на существование 3-раскраски за $\mathcal{O}\left(1.3478^n\right)$ с помощью другой **NP**-полной задачи (3,2)-SSS и покажем как его можно оптимизировать до $\mathcal{O}\left(1.3446^n\right)$.

Keywords
3-coloring, NP, SSS

1 Введение

Задача 3-раскрашиваемости заключается в проверке существования раскраски данного графа правильным образом в 3 цвета. Как известно, эта задача является **NP**-полной, а значит для ее решения не существует полиномиального алгоритма. Поэтому важно как можно сильнее уменьшить основание c показательной функции времени работы $\mathcal{O}\left(c^n\right)$.
Мы приведем алгоритм для $c = 1.3446$. Основная идея заключается в использовании обобщения 3-раскраски: (a,b)-SSS. В этой задаче дан набор из n вершин, которые надо раскрасить в a цветов так, чтобы выполнялись некоторые ограничения, каждое из которых запрещает раскраску кортежа из b вершин. Легко проверить, что 3-раскраска является (3,2)-SSS. Далее с помощью двойственности (a,b)-SSS и (b,a)-SSS и решения (2,3)-SSS решается проблема 3-раскраски.

1.1 Связи с другими работами и близкие результаты

Проблема 3-раскраски уже давно имеет несколько решений, использующих максимальные независимые множества, которые дают константы 1.4422 [1] и 1.415 [2]. Эта работа, описывающая алгоритм за 1.3446, заимствует большинство утверждений и идей из статьи Ричарда Бейгеля и Дэвида Эппштейна [3]. Они же в последствии усовершенствовали этот алгоритм, получив константу 1.3289 [4]. На данный момент это самый быстрый из известных алгоритмов.

2 Техническая часть

2.1 Определения

Введем все необходимые определения:

Определение 2.1. Задача 3-раскрашиваемости. Дан граф G. Требуется проверить, можно ли раскрасить его вершины в три цвета таким образом, чтобы никакие две одноцветные вершины не были соединены ребром.

Определение 2.2. Задача SSS (symbol-system satisfiability) - задача выполнимости системы символов. Даны:

- Набор вершин $\{v_1, \ldots, v_n\}$
- Для каждой вершины v_i список цветов, в которые ее можно покрасить $\{c_{i,1}, \ldots, c_{i,k_i}\}$
- Набор ограничений в виде кортежей пар из вершины и цвета $\{((v_1, c_1), \ldots, (v_m, c_m)), \ldots\}$

Говорим, что кортеж из набора ограничений выполняется, если все вершины из него покрашены в соответствующие цвета. Требуется проверить можно ли каждой вершине назначить один цвет из соответствующего ей списка таким образом, чтобы ни один кортеж не выполнялся.

Определение 2.3. Задача (a,b)-SSS - это SSS задача, в которой списки цветов вершин длины не больше a, длина каждого кортежа из набора ограничений не больше b.

2.2 Утверждения и леммы

Утверждение 2.1. *Задача 3-раскраски является (3,2)-SSS задачей.*

Доказательство. Список цветов для каждой вершины одинаков и состоит из трех цодинаковы ивестов: $C = \{1, 2, 3\}$. Рассмотрим следующие ограничения: $\{((v_1, c), (v_2, c)) \mid (v_1, v_2) \in E, c \in C\}$. Они и накладывают условие, запрещающее красить соседние вершины в один цвет, то есть условия задачи 3-раскрашиваемости. □

Лемма 2.2. *Каждую задачу из (a,b)-SSS можно свести к задаче из (b,a)-SSS*

Доказательство. Определим объекты двойственной задачи:

- Каждая вершина v'_i - кортеж из ограничений $((v_{k_{1,i}}, c_{k_{1,i},m_{1,i}}), \dots, (v_{k_{b,i}}, c_{k_{b,i},m_{b,i}}))$
- Цвета - пары (v_i, c_j). Набор цветов C'_i для вершины v'_i - множество пар из кортежа, соответствующего этой вершине. Поскольку размер кортежа в исходной задаче был не больше b, то и размер множества цветов для каждой вершины в двойственной задаче будет не больше b.
- Ограничения: набор кортежей вида $((v'_{k_1}, (v_i, c_{i,1})), (v'_{k_2}, (v_i, c_{i,2})), \dots, (v'_{k_a}, (v_i, c_{i,a})))$, где i от 1 до n (то есть v_i по всем вершинам исходной задачи), k_j пробегает индексы новых вершин, для которых: $(v_i, c_{i,j}) \in C'_{k_j}$ (то есть v'_{k_j} пробегает по всем вершинам новой задачи, набор цветов которых содержит соответствующий цвет, $(v_i, c_{i,j})$). Видно, что длина полученного кортежа не превосходит a, так как она равна мощности множества цветов в исходной задаче, которая не превышает a.

По построению ясно, что двойственная задача лежит в (b,a)-SSS.

Если для исходной задачи существовала раскраска, то это значит, что в каждом кортеже из ограничений существовала хотя бы одна пара (v, c), которая не выполнялась, в этот цвет и покрасим соответствующую вершину новой задачи (этот цвет лежит в наборе цветов вершины по построению). Также заметим, что для фиксированного i и произвольного l среди пар $(v_i, c_{i,l})$ хотя бы одна выполняется, следовательно в соответствующий ей цвет не будет покрашена ни одна новая вершина, поэтому ни одно новое ограничение не выполняется.

Теперь покажем следствие в обратную сторону: если существует решение новой задачи, то существует решение исходной. Зафиксируем i и для всех l рассмотрим набор цветов $(v_i, c_{i,l})$, если для каждого l нашлась вершина v'_{k_l}, покрашенная в этот цвет, то значит выполняется ограничение $((v'_{k_1}, (v_i, c_{i,1})), (v'_{k_2}, (v_i, c_{i,2})), \dots, (v'_{k_a}, (v_i, c_{i,a})))$. Следовательно $\forall i \exists l$ в цвет $(v_i, c_{i,l})$ не покрашена ни одна новая вершина. Тогда рассмотрим такую раскраску (вершину v_i в цвет $c_{i,l}$). Проверим, что не выполнено ни одно ограничение: если нашлось ограничение, в котором выполнена каждая пара, то в новой задаче соответствующая ему вершина покрашена в цвет какой-то пары из этого ограничения (v, c), но значит мы не могли покрасить вершину v в цвет c, так как брали пары, в которые не покрашена ни одна новая вершина. Противоречие. Значит ни одно ограничение не выполнено и раскраска исходной задачи существует. □

Лемма 2.3. *Если задача из (a,2)-SSS содержит вершину, которую разрешено красить лишь в два цвета, то она сводится к задаче из (a,2)-SSS с количеством вершин на 1 меньше.*

Доказательство. Пусть вершину v можно покрасить только в цвета R и G. Для каждой пары ограничений вида $((v, R), (u, A))$ и $((v, G), (w, B))$ добавим ограничение $((u, A), (w, B))$. И выкинем вершину v со всеми ограничениями, содержащими ее. Тогда если существовала правильная раскраска, то либо (v, G), либо (v, R) было верным, а значит (u, A) или (w, B) не должно выполняться, следовательно новые ограничения $((u, A), (w, B))$ не выполняются. И в другую сторону, предположим в новой задаче есть правильная раскраска, а в исходной нельзя докрасить никаким образом v, то есть существует ограничение выполняющееся ограничение для цвета R: $((v, R), (u, A))$ и цвета G: $((v, G), (w, B))$, но это значит, что выполняется и $((u, A), (w, B))$. Противоречие, так как это была правильная раскраска для новой задачи. □

Утверждение 2.4. *В задаче (a,b)-SSS ограничений не больше $(an)^b$, то есть полином.*

Доказательство. Каждое ограничение состоит из не более b пар, каждая пара выбирается an способами: вершина n способами, цвет a способами. В итоге получается $(an)^b$ □

Замечание 2.5. *По предыдущему утверждению операция из леммы 2.3 делается за полином.*

Далее будем работать только с задачами (3,2)-SSS, в которой каждую вершину можно покрасить в три цвета: R, G, B.

Лемма 2.6. *Если $((v, R), (w, R))$ единственное ограничение, содержащее (v, R), то ответ на задачу не изменится при добавлении 4 ограничений: $\{((v, C_v), (w, C_w)) : C_v \in \{B, G\}, C_w \in \{B, G\}\}$*

Доказательство. Понятно, что решение не могло появиться, так как мы только увеличиваем условия. Теперь предположим в исходной задаче была раскраска, в которой w красилась в R, тогда понятно, что ни одно новое ограничение не выполнено, так как $C_w \in \{B, G\}$. Если же w красилась в B или G, то перекрасим v в R, поскольку $((v, R), (w, R))$ единственное ограничение, содержащее (v, R), при таком перекрашивании ничего не сломается и новые ограничения не будут выполнены. □

Лемма 2.7. *Если были ограничения $((v, R), (w, G)), ((v, R), (w, B)), ((w, R), (x, R))$, то добавление ограничения $((v, R), (x, R))$ не изменит множество корректных раскрасок.*

Доказательство. Если v была покрашена в R, то для выполнимости первых двух ограничений w должна быть покрашена в R, тогда из третьего ограничения x в B или G, получается новое ограничение не выполняется. Если v покрашена не в R, то новое ограничение также не выполняется. □

3 Алгоритмы

3.1 Основной алгоритм

Теорема 3.1. *Задачу (3,2)-SSS можно решить за* $\mathcal{O}(1.38028^n)$

Доказательство. Пусть задача с n вершинами решается за $T(n)$. Рассмотрим все пары вида (v, R), для каждой пары рассмотрим множество всех ограничений, содержащих ее и разберем несколько случаев:

1. Нашлась пара, для которой эти ограничения содержат хотя бы три различных вершины (не включая вершину v из пары).

   - Покрасим вершину v в цвет R. Ограничения, содержащие (v, R) запрещают выполняться другим парам из них (так как эта пара выполнена), значит каждой вершине из этих пар запрещается краситься в один из цветов. По лемме 2.3 и замечанию 2.5 их можно выкинуть за полином. Поскольку таких вершин было хотя бы 3 и еще вершина v (для которой цвет определен), этот случай решается за $T(n-4)$.
   - Если для предыдущего варианта не нашлась раскраска, то v покрашена в один из двух оставшихся цветов. По лемме 2.3 ее можно удалить. Далее решаем за $T(n-1)$.

   В итоге получаем рекурренту $T(n) = T(n-1) + T(n-4)$. Будем искать решения вида $T(n) = \mathcal{O}(x^n)$. Тогда $x^n = x^{n-1} + x^{n-4} \Leftrightarrow x^4 - x^3 - 1 = 0$. $x = 1.380277\ldots$ является его решением, следовательно оценка $T(n) = \mathcal{O}(1.38028^n)$ подходит.

2. Нашлась пара, для которой это ограничение единственно: $((v, R), (w, R))$. Тут разберем еще несколько подслучаев:

   (a) Для пары (w, R) существует ограничение, содержащее ее и вершину отличную от v, w, то есть $\exists((w, R), (x, C)), x \neq v, w$. Далее действуем аналогично случаю 1:

   - w покрашена в R. Тогда x не в C и v не в R. По лемме 2.3 удаляем x и v. Получается $T(n-3)$.
   - w покрашена в B или G. Тогда покрасив v в R мы не изменим ответ на задачу, так как (v, R) встречается в единственном ограничении с (w, R), которое не выполнено. Удалив по лемме 2.3 w, получаем $T(n-2)$.

   Получилась рекуррента $T(n) = T(n-2) + T(n-3)$. Аналогично пункту 1, получаем $x^3 - x - 1 = 0 \Rightarrow x = 1.3247\ldots < 1.38028$ следовательно $T(n) = \mathcal{O}(1.38028^n)$ подходит.

   (b) Для пары (w, R) существует еще одно ограничение, однако в нем фигурирует v, то есть $((w, R), (v, C)), C \neq R$. Тогда если v можно покрасить в C, то ее можно покрасить и в R, так как если v в C, то w не в R, а значит единственное ограничение, содержащее (v, R): $((v, R), (w, R))$ уже не выполнено, то есть v можно перекрасить в R. Получается можно ограничить возможные цвета для v двумя (не C). По лемме 2.3 ее можно выкинуть. Задача свелась к $T(n-1)$.

   (c) Для пары (w, R) не существует других ограничений, содержащие ее. Рассмотрим 4 пары $\{(x, C) : x = v, w; C = B, G\}$, по лемме 2.6 добавим 4 ограничения (комбинации этих 4 пар). Заметим, что других ограничений, содержащих v и w нет. Разберем два подслучая:

   i. Для некоторой пары из этих четырех существует единственное ограничение, содержащее ее и вершину отличную от v, w. Пусть это ограничение $((v, B), (x, B)), x \neq v, w$.

   - x покрашена в B. Тогда v покрашена не в B и по лемме 2.3 удаляем ее. Получается $T(n-2)$.
   - x покрашена не в B. Тогда v можно покрасить в B, так как ограничение $((v, B), (x, B))$ единственное (кроме двух, содержащих w) содержит (v, B), и w в R, так как ограничение $((v, R), (w, R))$ единственное содержит (w, R) (поэтому два ограничения, содержащие (v, B) и w также не выполнены). По лемме 2.3 удаляем x. Получаем $T(n-3)$.

   В итоге получается рекуррента из предыдущего случая.

   ii. Для некоторой пары (допустим (v, B)) таких ограничений вообще нет, тогда красим v в B, w в R. Сводимся к $T(n-2)$

   iii. Для каждой из 4 пар существует хотя бы два таких ограничения. Возьмем, например, (v, B). Для нее есть два ограничения $((v, B), (x, G))$, $((v, B), (y, R))$, $x, y \neq v, w$. Если $x \neq y$, то вместе с $((v, B), (w, G/B))$ получится случай 1 для пары (v, B). Значит $x = y$, то есть $((v, B), (x, G))$, $((v, B), (x, R))$, $x \neq v, w$. Если для (x, B) нет ограничений, то красим x в B и получаем $T(n-1)$, значит есть $((x, B), (z, C)), z \neq x$. Тогда по лемме 2.7 можно добавить $((v, B), (z, C))$.

Если нашлось $z \neq v, w$, то для пары (v, B) применим случай 1. Если все $z = v$, то $C = B$ или G. Если $C = B$, то пара (v, B) в трех ограничениях с x всех цветов, значит v нельзя красить в B, по лемме 2.3 свелось к $T(n-1)$. Если $C = G$, то красим v в B, w в R, x в B (такая раскраска возможна, так как с парой (x, B) единственное ограничение $((x, B), (v, G))$, с парой (w, R) единственное ограничение $(w, R), (v, R)$, с парой (v, B) только ограничения, содержащие w и x, причем там нет (x, B) и (w, R)), значит задача свелась к $T(n-3)$. Остался вариант $z = w$, $C = B/G$, пусть $C = B$. В итоге есть следующие ограничения: $((v, B), (x, G))$, $((v, B), (x, R))$, $((x, B), (w, B))$. Поскольку для пары (w, B), как и для (v, B) существует 2 ограничения с вершинами отличными от v, w, то есть еще одно ограничение $((x, R/G), (w, B))$ (тут x, так как иначе будет случай 1), возьмем R. Получается $((v, B), (x, G))$, $((v, B), (x, R))$, $((x, B), (w, B))$, $((x, R), (w, B))$. Аналогичными рассуждениями для (v, G) получаем ограничения $((v, G), (y, C_1))$, $((v, G), (y, C_2))$, и для (w, G) получаем $((w, G), (z, C'_1))$, $((w, G), (z, C'_2))$. Ранее мы полагали $(z, C) = (w, B)$, сейчас мы можем для (v, G) либо тоже взять (w, B) и тогда будет $y = x$, либо (w, G), тогда будет $z = y$. В любом случае все пары $(v/w, G/B)$ в ограничениях максимум с двумя другими вершинами. Заметим, что либо v, либо w можно покрасить в R, но другую придется красить в B или G. Чтобы можно было покрасить v в B должно быть не выполнено (x, G) и (x, R), то есть x должен быть B, аналогично для трех других случаев получаем: x в G, y в C_3, z в C'_3. Достаточно выполнимости одного из этих условий, для этого не должно быть одновременно выполнено отрицание к каждому из них, отсюда несложно получаются ограничения на x, y, z, и поскольку две из них равны, размер ограничений не больше 2. В итоге свелось к $T(n-2)$ (выкинули v, w и добавили несколько ограничений).

3. Нашлась пара, для которой хотя бы три ограничения, но не выполнен первый случай (то есть все вместе они содержат не более двух различных вершин помимо v). Разберем несколько подслучаев:

   (a) Эти ограничения содержат лишь одну вершину отличную от v: $((v, R), (w, R))$, $((v, R), (w, G))$, $((v, R), (w, B))$. Тогда одна из трех правых пар точно выполнена, а значит v точно не R. Получается $T(n-1)$.

   (b) Ограничений ровно 3. Тогда, с учетом предыдущего подслучая, они такого вида: $((v, R), (x, R))$, $((v, R), (w, G))$, $((v, R), (w, B))$. Если (w, R) не содержится ни в одном ограничении, то красим w в R и получаем $T(n-1)$. Если содержится ровно в одном ограничении, то это случай 2. Получается существует два ограничения вида $((w, R), (z, C))$. По лемме 2.7 можно добавить ограничение $((v, R), (z, C))$. Значит $(z, C) \in \{(x, R), (v, B), (v, G)\}$, причем оно принимает два значения из этих трех. Рассмотрим два варианта раскраски:

      - w красим в R. Если (z, C) принимает два значения с x и v, то каждая из этих двух вершин может краситься лишь в два цвета. Получаем $T(n-3)$ по лемме 2.3. Если $(z, C) = (v, B), (v, G)$, то v можно покрасить только в R, значит x не в R. Получаем тоже $T(n-3)$.
      - w красим не в R. Тогда v не в R. Получается $T(n-2)$.

      В итоге получилась рекуррента $T(n) = T(n-2) + T(n-3)$. Она уже решалась выше.

   (c) Ограничений 4: $((v, R), (x, G))$, $((v, R), (x, B))$, $((v, R), (w, G))$, $((v, R), (w, B))$. Аналогично предыдущему варианту (w, R) и (x, R) содержатся только в ограничениях с $(v/x/w, G/B)$. Тогда можно покрасить v, w, x в R и оно не сломает другие ограничения.

4. Не выполнены предыдущие случае, то есть для всех пар ровно по два ограничения, содержащие их. Тогда можно построить граф, в котором вершины - пары, ребра - ограничения. В нем степень каждой вершины 2, значит он разбивается на объединение не пересекающихся циклов. Разберем несколько подслучаев:

   (a) Есть путь из пар длины хотя бы 5: $A - B - C - D - E$ и все 5 вершин в парах различны. Тогда возможны три взаимоисключающих варианта раскраски: красим пару B, тогда раскраска вершин из соседних пар возможна лишь в два цвета, то есть по лемме 2.3 получаем $T(n-3)$. Аналогично для пары C получаем $T(n-3)$. Далее красим пары A и D, получается два цвета для B, C, E, то есть $T(n-5)$. В итоге рекуррента $T(n) = 2T(n-3) + T(n-5)$. Решаем: $x^5 - 2x^2 - 1 = 0, x = 1.3639\ldots < 1.38028$.

(b) Есть квадрат $A-B-C-D$ из различных вершин в парах. Тогда его можно покрасить двумя вариантами: A и C или B и D. В обоих случаях можно удалить все 4 вершины. Получается $T(n) = 2T(n-4) \Rightarrow x^4 - 2 = 0 \Rightarrow x = 1.189\ldots < 1.38028$.

(c) Есть последовательность $A - B - C$, где вершина в A совпадает с вершиной в C. Пусть $A = (v, B), C = (v, G), B = (w, R)$. Тогда посмотрим на пару (v, R), она в двух ограничениях. Если среди них есть ограничение без w: $((v, R), (z, R))$, то по лемме 2.7 можно добавить ограничение $((w, R), (z, R))$. А это уже один из предыдущих случаев. Значит оба соседа пары (v, R) с вершиной w, это $(w, B), (w, G)$ ((w, R) уже использовалась). Тогда можно покрасить w в R и v в R, ведь их соседи (а это пары $(v/w, B/G)$ при такой раскраске не выполнены).

(d) Есть последовательность $A - B - C - D$, где вершина в A совпадает с вершиной в D, причем они не образуют треугольник. Здесь не могут быть одновременно выполнены A и D, а значит можно покрасить либо B, либо C в свой цвет. В обоих случаях после применения 2.3 получается $T(n-3)$, в итоге $T(n) = 2T(n-3)$, $x^3 = 2 \Rightarrow x = 1.259\ldots < 1.38028$.

(e) Есть последовательность $A - B - C - D - E$, где вершина в A совпадает с вершиной в E, но цвета не совпадают. Маленькими буквами будем обозначать вершины, тогда путь станет таким: $a-b-c-d-a$. Разберем подслучаи:

   i. Если после второй a идет не b, то это один из предыдущих вариантов (если не c, d, то пункт (a), если c, то пункт (d), если d, то пункт (c)).

   ii. После второй a идет b, далее по аналогичным причинам $c - d$, получается $a - b - c - d - a - b - c - d \sim A - B - C - D - E - F - G - H$. Если на этом цикл замкнулся, то максимум могут быть выполнены 3 из 8 пар. То есть одна вершина должна быть покрашена в третий цвет (которого нет в двух парах из этого цикла), а остальные 3 можно покрасить в цвета из цикла. Это значит, что оставшиеся 4 пары, содержащие эти вершины можно попарно соединить, так как будет выполнена лишь одна из них. Тогда получается один из предыдущих случаев.

   iii. Если цикл не замкнулся, то есть еще одна четверка $a - b - c - d$. Тогда получается эти 4 вершины встречаются только тут и они легко красятся.

(f) Весь граф разбился на треугольники. Рассмотрим двудольный граф: вершины одной доли соответствуют вершинам задачи, вершины второй доли - треугольникам из пар. Соединяем ребром, если вершина содержалась в одной из пар треугольника. Тогда степени вершин полученного двудольного графа = 3. Значит в нем есть паросочетание. В соответствии с ним и покрасим вершины. То есть в этом случае ответ всегда положительный.

$\square$

Замечание 3.2. *По утверждению 2.1 задача 3-раскраски является (3,2)-SSS (ограничения по графу строятся за полином, так как их всего полином по утверждению 2.4), а значит мы уже умеем решать ее за* $\mathcal{O}\left(1.38028^n\right)$.

3.2 Оптимизации

Оптимизация 3.3. *Пусть S - некоторое множество вершин графа G, который проверяется на 3-раскрашивоемость; Q - множество соседей S. Тогда для фиксированной раскраски множества S, каждая вершина Q красится не более, чем в два цвета (так как хотя бы один уже занят соседом из S), выкинем S и построим задачу (3,2)-SSS с соответствующими ограничениями на цвета. По лемме 2.3 можно выкинуть вершины Q, модифицировав ограничения. Получится задача (3,2)-SSS с $k = n - |S| - |Q|$ вершинами. Поскольку всевозможных раскрасок S $3^{|S|}$, получаем время* $\mathcal{O}\left(3^{|S|} \cdot 1.38028^k\right)$.

Далее основная идея найти небольшое множество S с большим числом соседей Q.

Для дальнейших рассуждений введем несколько определений и обозначений:
Пусть дан граф $G = (V, E)$. Пусть в нем каждая вершина степени хотя бы 3 (иначе удалим все вершины меньшей степени, так как их гарантированно можно раскрасить).

- X - максимальное по включению независимое подмножество V такое, что у любых двух вершин из него нет общего соседа.

- Y - множество всех соседей X. Заметим, что у каждой вершины X хотя бы 3 соседа в Y.

- $Z = V \setminus X \setminus Y$. Заметим, что каждая вершина из Z не соединена ни с одной вершиной из X (иначе ее можно добавить в Y) и соединена хотя бы с одной вершиной из Y (иначе ее можно добавить в X).

- F - некоторый лес деревьев с корнями в X, покрывающий G таким образом, что дети вершин из Y только в Z (то есть его глубина 3: корень из X, соседи корня из Y (их хотя бы 3), подмножество в Z)
- Поддерево F с корнем в вершине $v \in Y$ называется:
  - клубом, если у него нет детей в Z
  - палкой, если у него ровно один ребенок из Z
  - вилкой, если у него ровно два ребенка из Z
  - метлой, если хотя бы три ребенка из Z

Пусть S некоторое множество вершин, V множество не соседей S. При подсчете времени каждая вершина из S даст множитель 3, а каждая вершина V даст множитель 1.38028. Тогда определим *стоимость дерева* T как произведение множителей, которые дают его вершины. А *стоимость вершины* v из дерева T леса F как корень из стоимости дерева T степени количества вершин в дереве. Иными словами $cost(T) = 3^{|T \cap S|} * 1.38028^{|T \cap V|}, cost(v) = (cost(T))^{1/|T|}$. Время работы можно оценить как $\mathcal{O}\left(\max_v cost(v)^n\right)$, так как время равно произведению стоимостей деревьев леса F, а стоимость дерева оценивается $\max_v cost(v)^{|T|}$.

Лемма 3.4. *Пусть дерево T с тремя поддеревьями, хотя бы два из которых вилки или мётла. Тогда стоимость вершины из T не превосходит* $(6 \cdot 1.38028^2 + 3 \cdot 1.38028^3)^{1/10} \sim 1.3446$.

Доказательство. Разберем несколько случаев:

1. Все три поддерева мётла. Выберем за S корни всех трех поддеревьев (то есть $Y \cap T$). Тогда V для данного дерева пустое. То есть $cost(T) = 3^3$, $|T| \geqslant 1 + 3 + 9 = 13 \Rightarrow cost(v) \leqslant 3^{3/13} \sim 1.289$.

2. Есть поддерево, являющееся вилкой, палкой или клубом. Обозначим корни поддеревьев x, y, z и упорядочим их по количеству детей: x, y мётла или вилки (по условию), z - вилка, палка или клуб (по предположению). Пусть k - количество детей $z \Rightarrow k \in \{0, 1, 2\}$. Будем перебирать раскраски пары вершин x и y. В 6 случаях они разного цвета, тогда корень дерева красится однозначно и для z возможны лишь два цвета, получается в каждом таком случае в три цвета можно покрасить только детей z, то есть выходит 1.38028^k. В 3 случаях x и y одного цвета, тогда корень можно покрасить в два цвета, а z с его детьми в 3 цвета, то есть выходит 1.38028^{k+1}. Суммарно получается стоимость дерева $6 \cdot 1.38028^k + 3 \cdot 1.38028^{k+1}$, размер дерева хотя бы $1 + 3 + k + 2 \cdot 2 = k + 8$ а стоимость одной вершины $\left(6 \cdot 1.38028^k + 3 \cdot 1.38028^{k+1}\right)^{1/(k+8)}$. Максимум для $k \in \{0, 1, 2\}$ достигается в $k = 2$ и равен $(6 \cdot 1.38028^2 + 3 \cdot 1.38028^3)^{1/10} \sim 1.3446$..

□

Лемма 3.5. *Дерево T имеет хотя бы 4 поддерева. Тогда стоимость вершины из T не превышает* $\left(3 \cdot 1.38028^8\right)^{1/13} \sim 1.3267$.

Доказательство. Выберем корень и корни поддеревьев-мётел. Тогда пусть всего k мётел и m поддеревьев другого типа. Выбрана $k+1$ вершина, с ними не соседствуют только дети m вершин не мётел, пусть их $l \leqslant 2m$. Тогда стоимость дерева $3^{k+1} 1.38028^l$, всего вершин $\geqslant 1 + k + m + l + 3k$, значит стоимость вершины $\left(3^{k+1} 1.38028^l\right)^{1/(1+4k+m+l)}$. Далее ищем ее максимум: оценим m как $\frac{l}{2}$ и возьмем производную по k и l, ее 0 в единственной точке $(l, k) \sim (9.19, -3.69)$. Получается максимум в граничной точке $k = 0, m = 4, l = 8$, так как $m \geqslant 8$, то есть равен $\left(3 \cdot 1.38028^8\right)^{1/13} \sim 1.3267$. □

Лемма 3.6. *Дерево T имеет 3 поддерева и у корня не более 3 внуков. Тогда стоимость вершины T не превышает* $(3 \cdot 1.38028^3)^{1/7} \sim 1.3432$.

Доказательство.

Берем в S корень, его не соседи k вершин, $k \leqslant 3$. Получается $(3 \cdot 1.38028^k)^{1/(4+k)}$. Максимум достигается при $k = 3$ и равен: $(3 \cdot 1.38028^3)^{1/7} \sim 1.3432$ □

Лемма 3.7. *Дерево T имеет 3 поддерева и одно из них метла. Тогда стоимость вершины T не превышает* 1.3446

Доказательство. Если второе поддерево метла или вилка, то по лемме 3.4 стоимость не превосходит 1.3446. Иначе оба оставшихся поддерева палки или клубы, выберем за S одну вершину - корень метлы. Тогда у нее максимум 4 не соседа. Значит стоимость дерева $3 \cdot 1.38028^4$, его размер хотя бы 9. Значит стоимость вершины $(3 \cdot 1.38028^4)^{1/9} \sim 1.303$. □

Замечание 3.8. *Предыдущие леммы покрывают все случаи, кроме единственного случая, когда есть ровно 3 поддерева, два из которых палки и одно вилка (иначе выполнено условие одной из предыдущих лемм). Назовем такое дерево плохим.*

Лемма 3.9. *Дерево T имеет три ребенка и четыре внука. То стоимость вершины* $(3 * 1.38028^4)^{1/8} \sim 1.3478$ *- всего на* 0.0032 *больше необходимого.*

Доказательство. Выберем корень в S. Тогда его не соседи 4 внука, всего вершин 8, получаем $(3 * 1.38028^4)^{1/8} \sim 1.3478$. □

Лемма 3.10. *Можно выбрать описанный выше лес F так, что в нем ни одно дерево не имеет ровно две палки и одну вилку среди поддеревьев.*

Доказательство. Доказательство сводится к очень большому перебору, опишем его основную идею: Вначале выберем лес как-нибудь, как было описано в определении F. Далее будем его редактировать, увеличивая потенциал:

$$-c_1 n + c_2 A + c_3 B + c_4 C + c_5 D$$

где A - количество деревьев с четырьмя детьми, B - количество других деревьев, C - число вилок и метел, D - число не плохих деревьев. Коэффициенты выберем $c_1 > c_2 > c_3 > c_4 > c_5$. Поскольку A, B, C, D - целые числа не превышающие n, увеличиться этот потенциал может лишь полиномиальное число раз, а значит и время работы алгоритма будет полиномиальным (при условии, что на одно изменение тратится полиномиальное время).

Пусть x и y соседние вершины, но из разных деревьев. Тогда скажем, что x может *забрать* y, если y является внуком в своем дереве или сыном в дереве с тремя детьми, в котором есть еще хотя бы одна вилка или метла. Скажем, что x может *переместиться* к y, если y - корень дерева или вершина второго уровня (сын) и при добавлении x в это дерево появится новая вилка или не появится плохого дерева.

Далее рассматривается плохое дерево T и доказывается, что вышеописанными операциями можно от него избавиться. Это доказательство перебирает 16 случаев (если учитывать подслучаи, то 26). □

4 Отчет

4.1 Тесты

Правильная работа кода проверялась на всех графах с количеством вершин от 4 до 8 включительно (с точностью до изоморфизма) и на нескольких графах (около 30000) с 9 вершинами. Графы для тестов были взяты на users.cecs.anu.edu.au. И на нескольких случайных графов с большим числом вершин.

4.2 Анализ

Также для разных вероятностей p была исследована зависимость средней времени работы на выборке графов из $G(n,p)$ от числа вершин n. И было проведено сравнение с простейшим алгоритмом со временем в худшем случае $\mathcal{O}(3^n)$. На рисунке приведены графики некоторых из них. Можно заметить несколько закономерностей:

- Графы с отрицательным ответом обрабатываются намного дольше, чем с положительным. Это объясняется принципом работы алгоритма: чтобы дать отрицательный ответ, надо перебрать все случаи и получить на них ответ "НЕТ для положительного же ответа достаточно одного случая с ответом "ДА".
- Для $p \geqslant 0.3$ с некоторого момента почти нет положительных ответов, для маленьких же p наоборот нет отрицательных ответов. 2, 1
- Для очень маленьких $p \sim 0.01$, то есть для разреженных графов, алгоритм работает быстрее, чем для больших. Это объясняется тем, что во-первых, граф разбивается на несколько компонент связности и для них независимо считается ответ, во-вторых, в соответствии с алгоритмом вначале за полином удаляются все вершины степени не больше 2, а в разреженных графах их довольно много. 1
- Для $p \sim 0.3$ алгоритм дает наибольшую разницу по времени с тривиальным. Также в этом случае начиная с $n = 20$ время почти не меняется. 2
- Для больших p тривиальный в среднем оказывается быстрее нашего алгоритма. Так как в почти полных графах довольно вероятна встреча K_4, поэтому тривиальный алгоритм быстро доходит до вершины, которую нельзя покрасить ни в один из 3 цветов. 4
- Для всех p логарифм среднего времени работы растет медленнее, чем линейно от n, а значит практически в среднем оно не чистая экспонента.

5 Выводы

Удалось реализовать корректно и быстро работающий алгоритм проверки графа на 3-раскрашивоемость.

Верхняя оценка практически не достижима, так как она достигается при очень неудачном разбиении графа на деревья, когда все они с тремя детьми вилками или палками, то есть если степень вершин небольшая. Однако мы удаляем все вершины степени меньше 3, поэтому этот случай почти никогда не достигается.

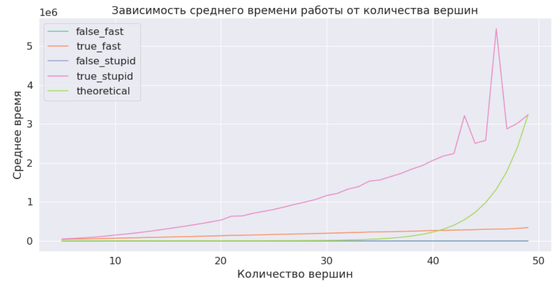
Алгоритм работает медленнее на графах с отрицательным ответом и большим числом ребер и быстрее на графах с положительным ответом и маленьким или средним числом ребер.

Исходный код и ноутбук с анализом лежат тут.

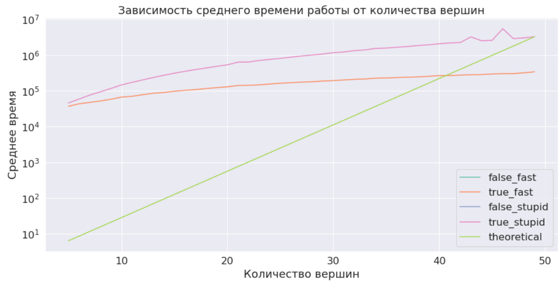
Список литературы

[1] Eugene L. Lawler. A note on the complexity of the chromatic number problem. *Inf. Process. Lett.*, 5:66–67, 1976.

[2] Ingo Schiermeyer. Deciding 3-colourability in less than $\mathcal{O}(1.415^n)$) steps. pages 177–188, 1994.

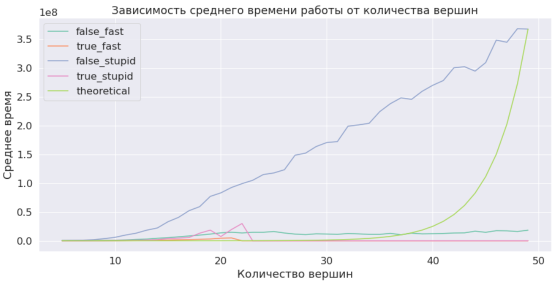


(a) Линейный масштаб

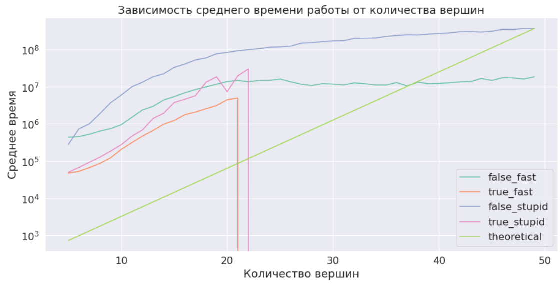


(b) Логарифмический масштаб

Рис. 1: $p = 0.01$

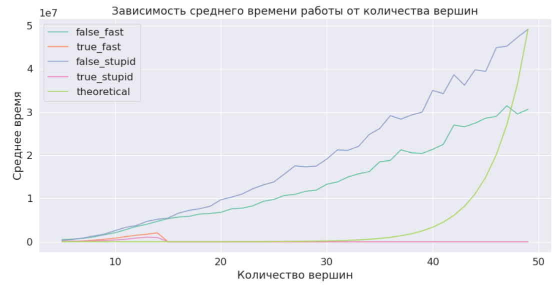


(a) Линейный масштаб

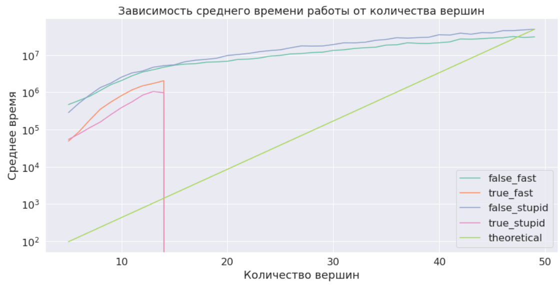


(b) Логарифмический масштаб

Рис. 2: $p = 0.3$

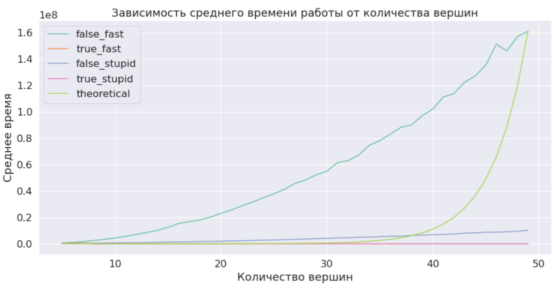


(a) Линейный масштаб

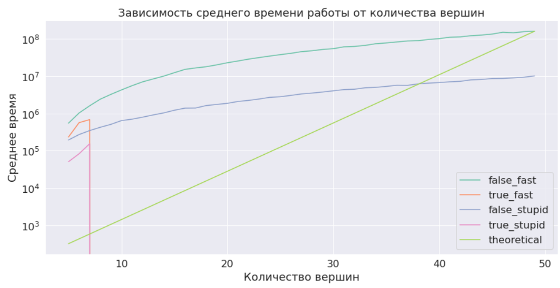


(b) Логарифмический масштаб

Рис. 3: $p = 0.5$



(a) Линейный масштаб



(b) Логарифмический масштаб

Рис. 4: $p = 0.9$

[3] R. Beigel and D. Eppstein. 3-coloring in time $\mathcal{O}\left(1.3446^n\right)$: a no-mis algorithm. page 444, oct 1995.

[4] Richard Beigel and David Eppstein. 3-coloring in time $\mathcal{O}\left(1.3289^n\right)$. *Journal of Algorithms*, 54(2):168–204, 2005.