Практическая работа по курсу «Сложность вычислений» на тему: «Раскраска 3-раскрашиваемого графа в $O(\sqrt{n})$ цветов»

Чиж Ярослав Владиславович
группа 694

1 Постановка задачи

Пусть дан граф G, вершины которого можно раскрасить в 3 цвета правильным образом, то есть одноцветные вершины не должны быть соединены ребром. Требуется построить полиномиальный алгоритм, который покрасит граф G правильным образом в $O(\sqrt{n})$ цветов.

2 Алгоритм

Пока в графе G есть вершина v степени хотя бы $\sqrt{n}$:

1. Красим вершину v в один цвет, ее соседей красим в два других цвета.
2. Удаляем вершину v и ее соседей из графа. На следующей итерации берем новые три цвета.

После данной процедуры остается H — подграф G, степени вершин в H строго меньше $\sqrt{n}$. Покрасим H в не более $\sqrt{n}$ цветов (при этом цвета в H будем брать отличными от цветов в $G \setminus H$):

1. На первом шаге покрасим одну вершину в 1-ый цвет.
2. На каждом следующем (k-ом) шаге будем красить k-ую вершину в минимально возможный цвет, то есть нужно просмотреть цвета всех соседей k-ой вершины и выбрать минимальный, который среди них не используется (минимальный во множестве натуральных чисел).

3 Доказательство корректности алгоритма

3.1 Покраска подграфа $G \setminus H$

Пусть v — вершина степени хотя бы $\sqrt{n}$, которую и соседей которой мы сейчас хотим покрасить. Понятно, что у v все соседи отличного от нее цвета. Надо доказать, что можно правильно раскрасить в два цвета соседей v. Предположим противное, пусть для правильной раскраски соседей v требуется хотя бы 3 цвета. Тогда добавим к ним вершину v. Чтобы покрасить v и ее соседей, потребуется хотя бы 4 цвета, ведь v с ними всеми соединена. Но это значит, что для покраски всего графа G потребуется хотя бы 4 цвета, а это противоречит предположению о том, что G — 3-раскрашиваемый граф.

Таким образом, каждая итерация покраски $G \setminus H$ корректна, а значит, и весь подграф $G \setminus H$ покрашен корректно, так как на каждой итерации берутся новые три цвета, то есть для вершин из различных итераций не может нарушиться правильность раскраски.

3.2 Покраска подграфа H

Понятно, что подграф H будет раскрашен правильно (просто следует из алгоритма), а значит, и весь граф G будет раскрашен правильно, потому что $G \setminus H$ раскрашен правильно (см. пункт 3.1), и цвета, используемые для H и $G \setminus H$, мы выбрали различными.

Надо только доказать, что на покраску H уйдет не более $\sqrt{n}$ цветов. Докажем это по индукции по количеству вершин m в подграфе H.

База для $m = 1$: красим вершину в один цвет, $1 \leqslant \sqrt{n}$ $(n \neq 0)$ — база верна.

Докажем переход индукции. Пусть мы уже покрасили k вершин в H правильным образом и хотим покрасить сейчас вершину $u \in H$. По алгоритму мы покрасим u, в минимально возможный цвет такой, которого нет среди ее соседей. Мы знаем, что соседей у u меньше $\sqrt{n}$, значит, среди цветов $1, \ldots, \sqrt{n}$ есть не занятый соседями u, следовательно, номер цвета для u будет не больше $\sqrt{n}$. То есть мы смогли покрасить $k+1$ вершину в H в не более $\sqrt{n}$ цветов — переход доказан.

Таким образом, на покраску H будет потрачено не более $\sqrt{n}$ цветов, что и требовалось доказать.

4 Подсчет количества цветов

Заметим, что на покраску $G \setminus H$ уйдет не более $3\sqrt{n}$ цветов, так как всего шагов с удалением вершины и ее соседей будет не больше $\frac{n}{\sqrt{n}} = \sqrt{n}$, а на каждом таком шаге используется не более трех цветов.

На покраску H, как уже говорилось выше, уйдет еще не более $\sqrt{n}$ — цветов.

Итого, на покраску всего графа уйдет не более $4\sqrt{n}$ цветов, что равно $O(\sqrt{n})$.

5 Асимптотика алгоритма

Реализация алгоритма приведена здесь.

Каждая итерация покраски и удаления вершин в $G \setminus H$ выполняется за $O(n^3)$, так как самая затратная часть по времени — это удаление вершин: каждое удаление занимает $O(n^2)$ времени. Итераций покраски не более $\sqrt{n}$, значит итоговая асимптотика для покраски $G \setminus H$: $O(n^3\sqrt{n})$.

На покраску H уйдет $O(n\sqrt{n})$ времени: в H не более n вершин, чтобы каждую покрасить, надо проверить соседей, которых меньше $\sqrt{n}$.

Итоговая асимптотика алгоритма по времени: $O(n^3\sqrt{n})$, что, конечно же, полиномиально от n, как и было заявлено в начале.

6 Результаты тестирования

Я протестировал алгоритм на всех 3-раскрашиваемых графах с количеством вершин от 1 до 7 (в которых между любыми двумя из трех компонент есть ребро; если это условие не выполенено, то граф уже точно является 2-раскрашиваемым), а также на случайных 3-раскрашиваемых графах с различным количеством вершин в трех компонентах (имеются в виду компоненты по цветам, см. подробности здесь) и с различными вероятностями появления ребра: 0, 0.01, 0.1, 0.5, 0.75, 0.95, 1. Тестирование прошло успешно: все тесты были пройдены, то есть граф в каждом случае был покрашен правильно, и количество цветов, в которые были раскрашены вершины графа, не превосходило заявленное количество: $4\sqrt{n}$.

Также отмечу, что на случайных графах (см. конкретные описания этих графов здесь) как при очень низкой вероятности появления ребра, так и при очень высокой, алгоритм красил граф в 1-10 цветов (чем больше вершин, обычно, тем больше цветов). А вот при средних вероятностях $0.1 - 0.75$ появления ребра алгоритм красил граф в более, чем 40 цветов (на 1200 вершинах), что хоть и не превосходит $4\sqrt{n}$, но всё же больше $\sqrt{n}$. Такие результаты обусловлены тем, что при крайне низких вероятностях появления ребра работает, в основном, вторая часть алгоритма (покраска подграфа H), которая является жадной и, действительно, хорошо работает при малых степенях вершин. При очень высокой вероятности появления ребра отрабатывает первая часть алгоритма несколько раз, а потом жадно красятся оставшиеся вершины, степени которых уже очень малы. При вероятностях не слишком больших и не слишком малых первая часть алгоритма почти не работает, по сути, почти весь граф красится второй частью алгоритма жадно, поэтому и требуется на покраску порядка $\sqrt{n}$ цветов.

7 Заключение

Мы реализовали полиномиальный алгоритм, который красит 3-раскрашиваемый граф правильным образом в $O(\sqrt{n})$ цветов и доказали его корректность.

Также, как мы поняли из пункта 5 (Асимптотика алгоритма), приведенный алгоритм будет довольно неплохо работать (имеется ввиду время работы) на графах, в которых степени вершин меньше $\sqrt{n}$ (n — количество вершин в графе): $O(n\sqrt{n})$ элементарных действий. Если же степени многих вершин не меньше $\sqrt{n}$, то время работы значительно увеличивается: $O(n^3\sqrt{n})$ — такая асимптотика вряд ли приемлема на практике, хотя это число полиномиально зависит от n.