

Проект по предмету «Сложность вычислений»

Реализация алгоритма проверки 3-раскрашиваемости графа (47-ой вариант)

В данной задаче нам требуется реализовать алгоритм проверки 3-раскрашиваемости графа. В данной статье мы опишем алгоритм работы нашей программы, его реализация, докажем его корректность, найдем время работы программы. Так же опишем, что дополнительно было написано в программе.

Описание алгоритма.

Найдем все максимальные по включению независимые множества в этом графе (множество вершин, в котором никакие две вершины из этого множества не соединены ребром, и при этом нельзя добавить еще вершину в это множество, чтобы оно осталось независимым). Не теряя общности, покрасим все эти вершины в первый цвет. Проверим можем ли мы покрасить все остальные вершины в оставшиеся два цвета. Если найдем хоть одну такую покраску, то ответ - «существует», иначе «не существует».

Реализация алгоритма.

а) нахождение максимальных независимых множеств :

Лемма. Либо вершина либо хотя бы одна из её соседей лежат в максимальном по включению независимом множестве.

Док-во. От противного. Пусть не лежит ни одна из соседей, тогда просто добавим эту вершину в множество.

Будем поддерживать некоторое множество, в котором в конечном результате будет находиться максимальное по включению независимое множество. Возьмем вершину с минимальной степенью. Тогда добавим в искомое множество либо эту вершину, либо одну из её соседей, переберем так для всевозможных способов, в конечном итоге на каждом шаге будем получать некоторое максимальное по включению независимое множество. Для остальных вершин нужно проверить, можем ли мы их покрасить в 2 цвета, а это решается обычной жадностью.

Корректность.

Мы красим некоторое подмножество в один цвет, пусть оно не является максимальным по включению независимым множеством, тогда просто добавим в него вершин, чтобы сделать его таковым, это, как минимум, не ухудшит возможность покраски оставшихся вершин в

остальные два цвета. В итоге, мы перебираем всевозможные максимальные по включению независимые множества, и для каждого из них пытаемся раскрасить граф => рассматриваем все случаи покраски, значит, алгоритм выполняется корректно.

Время работы.

Оценим число максимальных по включению независимых множеств. Заметим, что на каждом шаге можем брать некоторую вершину с минимальной степенью, тогда время работы алгоритма выглядит следующим образом. (пусть степень данной вершины равна d).

$T(n) \sim (d+1) * T(n - (d + 1))$, где $T(n)$ - время работы программы.

Тогда $T(n) = (d+1)^{n/(d+1)}$, легко доказать, что эта функция максимальна при $d = 2$ (например, по индукции, так же легко замечается, так как в итоге нужно найти минимум функции $k^{(1/k)}$).

Тогда у нас в итоге таких множеств не больше, чем $3^{(n/3)} \sim 1.44^n$.

Так же для каждого такого множества мы делаем 2-раскраску для оставшихся вершин, что занимает $O(n + m)$.

Итоговая сложность работы программы : $O(1.44^n * (n+m))$.

Дополнительная часть.

В программе так же было написано тестирование для программы.

а) Для проверки корректности работы программы, был так же написан алгоритм 3-раскрашиваемости графа за 3^n , и проверка совпадают ли ответы для данных алгоритмов. В результате проверки все ответы совпали.

б) Для проверки времени работы программы, был протестирован алгоритм на достаточно больших данных (для данной задачи), где N достигало порядка $\sim 30-40$. В результате проверки программа выдавала ответ в приемлимые для данной сложности время.