Хроматическое число

Базовые определения

Вершинная раскраска графа – присвоение вершинам графа цветов так, любые две смежные вершины имеют разные цвета.

Хроматическое число графа G – минимальное число цветов, в которое можно вершинно раскрасить данный граф **G**

Список смежности – словарь, в котором каждой метке вершины **v**, соответствует список смежной ей вершин **{u_1, …, u_n}**

Описание алгоритма

1. Строится список смежности **adjMap** для поданного на вход графа;
2. Строится массив **sortedVertices** меток вершин, отсортированный в порядке убывания количества соседей;
3. Создаётся хэш-таблица **vertexColors** для хранения соответствий вершина-цвет;
4. Для каждой вершины **v** из **sortedVertices** вычисляются цвета соседей $\{u_1, \ldots, u_k\}$. Далее подбирается цвет (начиная с 1) для вершины **v** так, чтобы $v \neq u_j$ для любого **j**. Найденный для **v** цвет сохраняется в **vertexColors** как пара **v: color** для быстрого доступа. Переменная **maxColor** обновляет своё значение, если **color > maxColor**.

Этот алгоритм является жадным.

Обоснование алгоритма

1. Если **|V| = 0**, то **ХЧ=0**
2. Если **|V| > 0**. Берётся минимальный цвет (**C=1**). Далее, если **C** есть в одной из соседних вершин, то **C** инкрементируется (**C = C + 1**) и так далее, пока не будет найден цвет, которого нет ни у одной соседней вершины. Это позволяет для каждой вершины выбирать минимальный возможный цвет.

Вершины с большей степенью имеют больше ограничений (больше соседей, которые уже могут быть раскрашены), поэтому их лучше раскрашивать раньше, пока доступно больше цветов. Это помогает уменьшить общее количество используемых цветов.

Контрпример для отсутствия сортировки вершин (дерево)

Обработка вершин в порядке увеличения числа соседей (**[6, 7, 3, 4, 5, 2, 1]**) даст неверный результат: **ХЧ=3**.

Если же отсортировать в порядке невозрастания числа соседей (**[1, 2, 6, 7, 3, 4, 5]**), то хроматическое число будет найдено корректно: **ХЧ=2**.