7.4. Дан неориентированный граф с максимальной степенью каждой вершины не больше трех. Разбить множество вершин на 2 доли так, чтобы у каждой вершины в в ее же доле было не больше одного соседа.

Для начала отметим, что задачу можно решать независимо по каждой компоненте связности, а значит необходимо описать алгоритм только для связного графа.

Итак, дан связный неориентированный граф с максимальной степенью каждой вершины 3. Выберем какую-нибудь вершину и сделаем из этой вершины DFS, одновременно с этим крася вершины в 2 цвета: 1 и 2. (начало аналогично первой задаче, оно же «построение остовного дерева = дерева поиску в глубину»). Как известно, дерево – двудольный граф, значит множество вершин можно разибить на две доли так, что у каждой вершины были соседи только в другой доле. Сделаем это.

Далее, как только мы закончили обрабатывать вершину, то есть цикл внутри DFS закончил свою работу и обработал все смежные вершины, скорректируем ее цвет: проверим все ребра, выходящие из этой вершины (их не более трех) и посмотрим на цвета.

Очевидно, что на всех смежных вершинах уже будут стоять какие-либо отметки (либо 1, либо 2): во-первых в эту вершину мы как-то попали, а поэтому родительская уже имеет цвет, во-вторых так как DFS закончил ее обрабатывать, значит все дочерние уже либо обработаны, либо помечены как used, а значит и покрашены.

Итак, посмотрим на смежные вершины и определим, плохая ли это вершина (то есть у нее в соседях 2 или 3 вершины одного с ней цвета). Заметим, что трех соседей одного цвета быть не может: так как мы в эту вершину как то попали, то по алгоритму покраски дерева у этой вершины и у ее родителя цвета разные. Значит единственной плохой ситуацией может быть та, когда обе ее смежные вершины (не считая «родительской») покрашены в ее же цвет. Тогда покрасим эту вершину в противоположный цвет (то фактически в цвет родителя). Это не приведет к поломке поддерева ниже этой вершины: оно уже просмотрено и там точно все хорошо. Сломаться все может только в родительской.

Рекурсивная часть DFS закончила свою работу и по стеку возвращаемся в родительскую вершину. Опять же, повторяем все шаги: смотрим на все ее смежные вершины, и, если надо, перекрашиваем ее в другой цвет.

Получается, что мы «поднимаемся» по дереву, исправляя ошибки на нижних уровнях и поднимая возможную ошибку на самый верх. После разворачивания последней итерации рекурсии (то есть закончив обрабаывать самую первую вершину, то есть корень), плохая ситуация может быть только в нем: все поддерево гарантированно корректно покрашено. Посмотрим что может быть в корне:

- Если корень первой степени, то все хорошо, очевидно.
- Если корень второй степени, и оба ребенка того же цвета, что и он – просто перекрасим его в другой цвет и все будет хорошо.
- Если корень третьей степени. Пусть корень имеет цвет 1. Все возможные 8 раскрасок его смежных вершин:
    - 111 – красим корень в цвет 2
    - 112 – красим корень в цвет 2
    - 121– красим корень в цвет 2
    - 122 – все уже хорошо
    - 211 – красим корень в цвет 2
    - 212 – все уже хорошо
    - 221 – все уже хорошо
    - 222 – все уже хорошо

Итак, в любой ситуации с раскраской корня, можно его перекрасить так, чтобы получить корректную раскраску дерева.