Попова Вероника Б9124-02.03.01сцт3

Алгоритм

Нам надо раскрасить сетку размером m на n в три цвета (красный, зелёный, синий), но так, чтобы две соседние клетки не были одинакового цвета. Важно учитывать, что это условие не только распространяется на столбцы, но и на строки. Я решила сначала находить правильные столбцы, проверять их парами на совместимость и потом уже соединять.

Обозначила цвета цифрами: 0, 1 и 2. То есть, по сути, столбец - это набор m чисел. Сделаем тогда это с помощью троичной системы, поэтому дальше мы будем делить на 3, чтобы убрать последнюю цифру, а %3, чтобы узнать какой последний цвет там лежит. Каждый столбец - цифра в троичной системе.

Для того, чтобы найти правильные столбцы, код перебирает всевозможные варианты столбца (это переменная total). Проверяем текущий цвет и предыдущий, если совпадают, то ломаем этот набор и начинаем новую последовательность. В этом участке кода использовала флаг skip, чтобы убрать ненужные наборы.

Для того, чтобы найти подходящие друг к другу столбцы, надо будет проверять с последней цифры каждого из двух столбцов, так как у нас троичная система. Если не подходит второй столбец, то ставим следующий. Если подходит, то мы записываем в двумерный массив совместимости g(sz, vector<bool>(sz, false)).

vector<long long> old(sz, 1) – сколько способов для раскрашивания предыдущих столбцов (начинаем с одного, так как первый столбец может быть любым).

vector<long long> cur(sz, 0) – сколько способов раскрасить столбцы, добавив новый.

Для того, чтобы всё соединить, мы проходимся по массиву g и добавляем столбцы, которые подходят. Для каждого нового столбца (j) мы складываем предыдущие способы из old[j] для всех i, где g[i][j] = true, потом записываем сумму в cur. После этого, old становится cur, а cur обнуляется.

В конце концов, как код прошёл все состояния, в old остаётся количество способов, только осталось их сложить и взять по модулю $10^9 + 7$.

Временная сложность

Создаём каждый столбец для проверки: перебираем числа от 0 до 3^m, а потом проверяем m цифр. Это получается $O(3^m * m)$, максимум 1215 операций, так как m <= 5.

Создаём массив совместимости: sz - количество подходящих столбцов, тогда максимальное количество при m <= 5 будет равен 3*2*2*2*2 = 48. Так как у нас пара столбцов рассматривается, то sz*sz, а для каждой из них m строк. Тогда получается сложность O(sz * sz * m) = 11520 операций.

Собираем все столбцы: каждый столбец сопоставляем с парой. У нас n <= 1000. Это получается: O(sz * sz * n) = 2304000 операций.

Общая сложность: $O(3^m * m + sz * sz * m + sz * sz * n) \approx O(n)$

Затраты памяти

Допустимые столбцы в массиве (int): O(sz) = 48 * 4 = 192 байта

Матрица совместимости (bool): O(sz * sz) = 2304 байта

Массивы для соединения (long long): O(sz * 2) = 48 * 2 * 8 = 768 байт

Общее количество затраченной памяти для максимального случая: 3,3 Кб