Задача A. И-го-го!

Задача A. И-го-го! Суть: Определить, может ли шахматный конь посетить все клетки доски $n \times m$, имея возможность заходить в клетки несколько раз.

Подзадача 2 ($n = 1$): На доске $1 \times m$ конь не может совершить ни одного хода, так как для хода буквой «Г» нужно как минимум 2 строки. Согласно примеру, при 1×1 ответ «YES», а при других значениях m конь не сдвинется. Однако, так как можно начинать с любой клетки и нужно посетить все, для $1 \times m$ при $m > 1$ это невозможно.

Подзадача 3 ($n = 2$): На доске $2 \times m$ конь может прыгать только в клетки через один столбец. Доска разбивается на две несвязные группы клеток (как «черные» и «белые» клетки, до которых нельзя добраться). Посетить все клетки невозможно.

Подзадачи 1 и 4 ($n, m > 4$ и общие случаи): Для большинства досок $n, m \geqslant 3$ конь может посетить все клетки (связный граф). Исключения составляют очень маленькие доски (например, 3×3, где центральная клетка недостижима).

Задача B. Круговорот сока в природе

Подзадача 1 ($n \leqslant 10$): Можно использовать простой перебор всех возможных путей из каждой вершины.

Подзадача 2 ($n \leqslant 1000$): Для каждой вершины запускаем процесс обхода, пока не встретим повторение. Запоминаем длину найденного цикла.

Подзадачи 3 и 4 ($n \leqslant 500000$): Требуется эффективный поиск циклов за $O(n)$. Поскольку из каждой вершины выходит только одно ребро, граф состоит из набора компонент, каждая из которых — это дерево, входящее в цикл.

1. Находим все циклы, используя поиск в глубину (DFS) или подсчет входящих степеней (удаление вершин с нулевой степенью входа, пока не останутся только циклы).

2. Выбираем цикл с максимальным количеством вершин.

Задача C. Далекое королевство

Суть: Заменить числа в матрице на минимально возможные натуральные числа, сохранив относительный порядок (больше, меньше, равно) в каждой строке и каждом столбце.

Подзадача 1 ($n = 1$): Это классическая задача на координатное сжатие одной строки. Сортируем уникальные элементы и заменяем их на их ранги (1, 2, 3...).

Подзадача 2 и 3 ($n, m \leqslant 100$):Построим граф зависимостей: если в какой-то строке(столбце) один элемент меньше другого, проводим ориентированное ребро (из меньшего в больший). Решением будет поиск самого длинного пути в полученном DAG (направленном ациклическом графе).

Подзадача 4 (Общий случай $n \times m \leqslant 1000000$):

Группируем равные элементы в строках и столбцах (используя систему непересекающихся множеств — DSU), так как они должны получить одинаковые значения.

Строим граф зависимостей между группами: если в строке/столбце одно число больше другого, добавляем ребро.

Значение группы — это длина максимального пути в графе до неё.

Задача D. Луг

Суть: Найти количество прямоугольных подобластей, содержащих хотя бы по одному цветку каждого вида, присутствующего на всем поле n×n.

Подзадачи 1 и 2 ($n \leqslant 20$): Полный перебор всех прямоугольников ($O(n^4)$) и проверка состава цветов в каждом.

Подзадача 3 и 4 ($n \leqslant 100$): Оптимизация проверки. Можно использовать префиксные суммы для каждого вида цветов (всего их до 26 — латинские буквы), чтобы за O(1) проверять наличие вида в прямоугольнике.

Подзадачи 5 и 6 ($n \leqslant 500$): Используется метод «двух указателей» или скользящего окна.

Фиксируем верхнюю и нижнюю границы прямоугольника (две строки).

Идем по столбцам, расширяя правую границу, пока не соберем все виды цветов.

Для каждой левой границы находим минимальную правую границу. Все прямоугольники правее этой границы также будут подходить.