

Задача из Озёрска

Рассмотрим случай, когда $n = 1$, то есть мы можем двигаться только в пределах одной плитки. Представим поле из $H \times W$ белых и чёрных клеток в виде неориентированного графа. Вершинами в нём будут белые клетки, а рёбра проведём между соседними клетками. В таком графе выделим компоненты связности. Пусть $comp_{r,c}$ — номер компоненты связности клетки (r, c) . Для ответа на запрос будет достаточно сравнить $comp_{R_1, C_1}$ и $comp_{R_2, C_2}$.

При переходе к $n > 1$ добавляются переходы между плитками. Пронумеруем плитки слева направо числами от 1 до N . Пусть клетка (R_1, C_1) лежит в плитке G_1 , а клетка (R_2, C_2) — в плитке G_2 . Если $G_1 > G_2$, то ответ «No», так как зазор между плитками можно проходить только в направлении слева направо. Если $G_1 = G_2$, то в этом случае мы можем двигаться только в пределах одной плитки, и здесь будет достаточно решения из случая $n = 1$.

Рассмотрим переход через зазор между двумя плитками. Мы попадаем из самого правого столбца одной плитки в самый левый столбец второй плитки. Пусть переход происходит в строке R . Тогда мы попадаем из компоненты связности $comp_{R, H-1}$ в компоненту связности $comp_{R, 0}$, если обе клетки $(R, H-1)$ и $(R, 0)$ — белые. На самом деле, задача свелась к тому, чтобы определить, можно ли ровно за $G_2 - G_1$ переходов через зазор между двумя плитками попасть из $comp_{R_1, C_1}$ в $comp_{R_2, C_2}$.

Построим ориентированный граф, вершинами в котором будут компоненты связности изначального графа, а рёбра проведём в соответствии с переходами через зазор между двумя плитками. В таком графе $\mathcal{O}(H)$ вершин будут иметь хотя бы одно входящее или исходящее ребро, а рёбер будет не больше H , так как для каждой строки плитки мы проводим максимум одно ребро.

Рассмотрим подробнее, как выглядит этот ориентированный граф. Во-первых, в нём могут быть петли, что равнозначно возможности сколь угодно долго оставаться в одной и той же компоненте связности при переходе между плитками. Во-вторых, подумаем, могут ли в этом графе существовать циклы, не являющиеся петлями. Докажем от противного, что граф ацикличен. Пусть цикл существует. Рассмотрим рёбра, лежащие на этом цикле. Каждому ребру соответствует строка, в которой осуществляется переход между вершинами. Каждой строке соответствуют две компоненты, между которыми проведено ребро. Выпишем сверху вниз номера компонент из самого левого столбца, входящих в цикл, — получим список a_i длины k . То же самое сделаем для самого правого столбца — получим список b_i длины k . Множества вершин в этих списках совпадают, так как они образуют цикл, и каждая вершина встречается там лишь единожды. Если для всех $1 \leq i \leq k$ выполняется $a_i = b_i$, то вершины цикл не образуют, так как каждое ребро — петля. Иначе существуют такие вершины x и y , что $pos_a(x) < pos_a(y)$ и $pos_b(x) > pos_b(y)$, где $pos_a(v)$ — позиция вершины v в списке a и $pos_b(v)$ — позиция вершины v в списке b . В таком случае компоненты, соответствующие вершинам x и y , обязаны пересечься хотя бы в одной клетке плитки, что противоречит тому, что это две разные компоненты связности.

Таким образом, доказали, что граф является DAG'ом с петлями (DAG — Directed Acyclic Graph — Ориентированный Ациклический Граф). В нём нужно узнавать, существует ли путь из одной вершины в другую, проходящий ровно по заданному количеству рёбер. Сначала рассмотрим случай, если в графе отсутствуют петли. Пусть $dp[x][z][k]$ — существует ли путь из x в z , проходящий ровно по k рёбрам. $dp[x][z][1] = 1$, если существует ребро из x в z . Для вычисления значений динамики при $k > 1$ переберём первое ребро на пути из x в z . Пусть оно ведёт в вершину y . Тогда $dp[x][z][k] = 1$, если нашёлся хотя бы один y , для которого $dp[y][z][k-1] = 1$. Заметим, что перебирать значения $k > H$ не имеет смысла, так как в графе не больше H вершин, и он ацикличен. Пересчёт всех состояний динамики можно реализовать при помощи обхода в глубину. Это будет работать за $\mathcal{O}(H^3)$, так как для фиксированных z и k мы перебираем пары (x, y) , каждая из которых соответствует ребру в графе, которых не больше H .

Осталось разобраться, что делать с петлями и каким образом они влияют на ответ. На самом деле, они позволяют пройти часть пути из начальной вершины, потом отходить необходимое количество рёбер по петле и затем добраться до конечной вершины. Пусть $inf[x][z]$ — существует ли путь из x в z , проходящий хотя бы по одной вершине, имеющей петлю. Вычисление значений этой динамики можно осуществить при помощи того же обхода в глубину. Пусть путь из x в z существует, а d — минимальное, такое что $dp[x][z][d] = 1$. Тогда, если $inf[x][z] = 1$, то из x в z существуют пути всех длин, больших либо равных d . Таким образом, при посчитанных значениях dp и inf можно

отвечать на запрос за $\mathcal{O}(1)$, и общее время работы программы будет $\mathcal{O}(H^3 + Q)$.