

---

## Разбор задачи «Древнее заклинание»

Автор задачи:

Николай Будин

Подготовка условия, решения и тестов:

Николай Будин

Автор разбора:

Николай Будин

Давайте рассмотрим граф с  $n \cdot m \cdot l$  вершинами. Будем обозначать вершину тройкой  $(x, y, z)$  ( $1 \leq x \leq n$ ;  $1 \leq y \leq m$ ;  $1 \leq z \leq l$ ), она будет соответствовать одновременно ячейке таблицы  $(x, y)$  и  $z$ -му символу заклинания.

Рассмотрим теперь, как построить ребра. Обозначим как  $f[x][y]$  символ в ячейке  $(x, y)$  таблицы, а  $s[z]$  — символ заклинания на позиции  $z$ . Пусть ячейки  $(x_1, y_1)$  и  $(x_2, y_2)$  являются соседними по стороне в таблице. Тогда проведем ориентированное ребро из  $(x_1, y_1, z)$  в  $(x_2, y_2, (z \bmod l) + 1)$ , если  $f[x_1][y_1] = s[z]$  и  $f[x_2][y_2] = s[(z \bmod l) + 1]$ .

Несложно заметить, что любой цикл в таком графе соответствует искомой последовательности. Осталось найти цикл или выяснить, что его нет. Это можно сделать с помощью обхода в глубину за время, пропорциональное сумме числа вершин и ребер нашего графа, то есть  $O(nml)$ .

Сделаем напоследок несколько технических замечаний. Поскольку в нашем графе в худшем случае около 8 миллионов вершин и 32 миллионов ребер, то явно строить граф и хранить списками смежности все ребра может не получиться из-за ограничений по памяти. Даже если аккуратно реализовать хранение ребер, размер графа будет существенно превышать размер кеша процессора. Работа с таким графом будет происходить медленно из-за постоянных промахов кеша, поэтому возможно превышение ограничения по времени. Эффективнее во время выполнения обхода в глубину, находясь в вершине  $(x, y, z)$ , перебирать 4 соседние клетки и проверять, можно ли в них перейти.