

# Кибер-взлом

*Автор задачи и разработчик: Николай Будин*

Чтобы решить третью подзадачу, можно было заметить, что каждому из игроков выгодно, независимо от действий второго, ходить так, чтобы он смог сделать как можно больше ходов. Для каждой вершины можно вычислить  $len[v]$  — длину самого длинного пути, начинающегося в вершине  $v$ . Для всех вершин, из которых достижим цикл,  $len[v] = \infty$ , после чего можно удалить все циклы, граф станет ациклическим, можно будет построить его топологическую сортировку и вычислить значения для оставшихся вершин с помощью метода динамического программирования. Пара вершин  $v, u$  входит в ответ, если  $len[v] > len[u]$ .

Для решения оставшихся подзадач, нужно было заметить, что в этой задаче был описан процесс, который является несимметричной комбинаторной игрой. Поймем, какие у этой игры состояния. Перед тем, как ход сделает первый игрок, нам важна только информация о позициях токенов первого и второго игроков. Перед ходом второго игрока, нам помимо позиций токенов также нужно знать символ, по которому перешел первый игрок на последнем ходу.

Можно построить граф, в котором вершины соответствуют описанным состояниям, а ребра соответствуют ходам игроков и всегда ведут из состояния первого типа в состояние второго и наоборот. Состояние, из которого нет ни одного перехода, является проигрышным. Если из состояния есть переход в проигрышное, то оно выигрышное. Если из состояния все переходы ведут в выигрышные, то оно проигрышное. Ответом является количество выигрышных состояний первого типа.

Таким образом, если бы граф состояний был ациклическим, можно было бы сделать его топологическую сортировку и для каждого состояния вычислить выигрышность/проигрышность с помощью метода динамического программирования. Можно доказать, что если исходный граф является ациклическим, то и граф состояний является ациклическим. Поэтому, это позволяет решить четвертую подзадачу. Для ускорения, можно не строить граф явно и воспользоваться ленивым динамическим программированием.

Если же граф состояний содержит циклы, то некоторые состояния являются ничейными, это означает, что игра из этого состояния никогда не завершится. Для того, чтобы определить тип каждого из состояний в циклической игре, можно воспользоваться стандартным методом — ретроспективным анализом. Его суть заключается в том, что он старается определить типы вершин начиная с тех, из которых нет переходов (напомним, что эти вершины являются проигрышными). Если в какой-то момент из вершины, тип которой мы еще не определили, появляется ребро в проигрышную, то эту вершину мы помечаем выигрышной. Если в какой-то момент из вершины, тип которой мы еще не определили, все ребра начинают вести в выигрышные, вершина становится проигрышной. Если таких вершин не осталось, процесс прерывается. Все вершины, тип которых мы не определили, помечаются ничейными.