Тураев Тимур, 504 (SE)
20.10.2013

7.3. Игра на дереве.

Мы помечаем вершины единицей, если, попав в эту вершину, в ней гарантировано выиграет первый. И двойкой - если второй. Как только обход в глубину выходит из вершины, это значит, что просмотрены все его дочерние вершины, а значит в них проставлены цифры один или два. Значит, можем определить кто выиграет в этой вершине: если сейчас ход первого (то есть вершина находится на четном расстоянии от корня), и есть хотя бы одна дочерняя вершина, помеченная единицей, то первый туда и пойдет, чтобы выиграть, значит помечаем нашу вершину единицей. Если ход первого и среди дочерних вершин нет единиц (то есть все двойки), то выиграет второй. И аналогично для второго.

Если же дочерних вершин нет (то есть мы в листе дерева), то правила такие: если сейчас ход первого, то он проиграл, помечаем лист "2" (выиграл второй), иначе - выиграл первый.

7.5. Бочки с водой.

Да, непонятно зачем трехмерный, если состояние однозначно определяется объемами первой и второй бочек. Заведем двумерный массив (a+1)*(b+1) и заполним его нулями.

Сделаем обход в глубину. Алгоритм его такой:

- used[v] = 1. Помечаем текущую вершину как использованную. В нашем случае это будет запись единицы в соответствующую ячейку массива.
- for all u in adj(v): по всем смежным вершинам. В нашем случае это перебор шести состояний (1<->2, 2<->3, 1<->3) и проверка, существует ли такое переливание вообще (не существует, если источник пуст или заполнен приемник)
- if not used[u]: проверяем число в двумерном массиве - идем только в неиспользованное состояние, то есть должен стоять ноль.
- dfs(u): запускаем алгоритм рекурсивно.

Закончим работу когда встретим нужный нам объем. Вывести ответ можно по-разному, можно сразу в DFS печатать состояние, если в нем достигнут ответ (и возвращать true), можно в тот двумерный массив передавать значение "родителя" и потом пройти по родителям и напечатать последовательность, приводящую к ответу.