

## Деревянный Морти

Задача состоит в проверке, имеет ли первое дерево подграф, изоморфный второму дереву. Давайте попробуем решить задачу с любой асимптотикой, а затем уберем все лишнее из решения.

Сделаем первое дерево корневым и решим задачу с помощью динамического программирования на его поддеревьях. Пусть  $\text{dp}[u][x \rightarrow v]$ , где  $u$  — вершина первого дерева, а  $x \rightarrow v$  — ориентированное ребро второго дерева, будет означать, возможно ли сопоставить вершины  $u$  и  $v$ , при этом

- сопоставляя поддерево  $u$  с поддеревом  $v$ ;
- при условии, что во втором дереве родитель  $v$  — это  $x$ .

Иными словами, мы сопоставляем ребра  $p(u) \rightarrow u$  и  $x \rightarrow v$ , а также поддеревья  $u$  и  $v$ .

Для подсчета такой динамики нам нужны значения всех  $\text{dp}[u'][v \rightarrow v']$ , где  $u'$  — сын  $u$ , а  $v'$  — сын  $v$ . Сам пересчет является проверкой, есть ли паросочетание в двудольном графе, которое покрывает левую часть — действительно, поддеревья каких-то детей  $u$  надо «покрыть» поддеревьями детей  $v$ , а всех остальных — добавить отдельно.

К сожалению, такая динамика работает очень медленно и не дает ответа на задачу. Давайте исправим обе эти проблемы сразу. Посмотрим на пересчет всех состояний динамики вида  $\text{dp}[u][x \rightarrow v]$  для разных  $x$ . Для их пересчета мы ищем совпадение для графов с одинаковой правой частью (все дети  $u$ ) и левой частью, которая отличается одной вершиной (все соседи  $v$ , кроме  $x$ ).

Вместо этого давайте сразу запустим поиск паросочетания на графе со всеми интересующими нас вершинами, то есть только один раз для пары вершин  $u$  и  $v$ . Тогда:

- если мы нашли паросочетание, которое покрывает левую часть, то ответ для всех  $x$  будет **true**;
- если мы не смогли покрыть хотя бы две вершины, то значение всех  $\text{dp}[u][x \rightarrow v]$  — это **false**;
- а если мы смогли покрыть все вершины, кроме одной  $x'$ , то для всех остальных  $x$  ответ **true**, а для  $x'$  мы запустим один **dfs** от нее, ищущий дополняющую чередующуюся цепь, и определим ответ для  $\text{dp}[u][x' \rightarrow v]$ .

Решение заканчивается здесь, для ответа остается только рассмотреть какой-то лист в качестве  $x$  и найти, есть ли соответствующий **dp**, равный **true**.

Остается только оценить его время выполнения — мы запускаем один алгоритм поиска паросочетания для каждой пары вершин  $(u, v)$ . Давайте посчитаем максимально возможное количество ребер во всех полученных графах, то есть  $\sum_{u,v} \deg(u) \cdot \deg(v) = \sum_u \deg(u) \cdot 2 \cdot m = 4nm$ .

Если честно оценить алгоритм Куна, то мы получим асимптотику всего решения  $\mathcal{O}(nm^2)$ , однако на конкретных графах алгоритм Куна на самом деле работает гораздо быстрее, поэтому такое решение уже проходит при достаточно аккуратной реализации. Если вы действительно хотите написать асимптотически более быстрое решение, напишите алгоритм Диница, чтобы получить решение за  $\mathcal{O}(nm\sqrt{m})$ .