
Разбор задачи «Энергия»

В задаче требовалось найти в дереве k непересекающихся путей так, чтобы сумма весов всех вершин на путях была максимальна.

Решение для маленьких k

В первых четырех подгруппах, при небольших значениях k , достаточно было перебрать путь в дереве, либо же два пути, и выбрать наилучший ответ. В подзадаче 4 требовалось оптимизировать этот перебор, запомнив для каждого поддерева максимальный вес пути в нем.

Решение для маленьких n

В подзадаче 5 достаточно было написать перебор.

Можно перебрать множество вершин, которые войдут в ответ, и проверить с помощью динамического программирования, что это множество вершин можно разбить на k путей.

Можно реализовать рекурсивный перебор, перебрав все возможные множества из k непересекающихся путей. Этот перебор необходимо реализовать аккуратно, чтобы не рассматривать по несколько раз одни и те же множества путей.

Полное решение

Для полного решения этой задачи воспользуемся динамическим программированием по поддеревьям.

Подвесим дерево за произвольную вершину, и посчитаем динамику. Состоянием динамики будет пара чисел (u, s) — вершина, поддерево которой мы рассматриваем, и количество путей, которые мы уже выбрали в этом поддереве. Для пересчета этой динамики в состояние также необходимо добавить булевый параметр $flag$ — верно ли, что вершина u является концом какого-то из взятых путей. Значением динамики будет искомый ответ. Таким образом, $dp[u][s][flag]$ — это максимальная сумма, которую мы можем получить, взяв из поддерева вершины u ровно s путей, при этом, если $flag = 1$, вершина u должна являться концом одного из путей.

Для пересчета этой динамики в каждой вершине будем считать вспомогательную динамику на префиксах ее детей. Пусть $dp'[u][p][i][flag]$ — это максимальная сумма, которую можно набрать, если рассматривать только поддеревья первых p сыновей вершины u . При этом параметр $flag$ может принимать одно из трех значений: $flag = 0$ означает, что вершина u не была задействована ни в одном из путей, $flag = 1$ означает, что вершина u задействована в одном из путей и является его концом, а $flag = 2$ означает, что вершина u задействована в одном из путей, но не является его концом.

Пересчитывать динамику dp' можно через значения dp для очередного сына каждой вершины. Достаточно лишь перебрать, сколько путей мы возьмем из поддерева сына, а также, что случится с путем, содержащим этого сына: он может либо продлиться вверх на один, либо склеиться с каким-то другим путем. Каждый из этих случаев дает нам переход в свое состояние динамики dp' .

Ответ на задачу будет равен $\min(dp[root][k][0], dp[root][k][1])$, где $root$ — корень дерева.

Оценим время работы данного решения. Всего есть $O(nk)$ состояний динамики, всего переходов от сына к отцу будет сделано $O(n)$, так в дереве $n - 1$ ребро. При каждом из переходов нам нужно перебрать, сколько путей мы берем из поддерева сына, а также сколько путей уже было взято. Итоговая оценка времени работы составляет $O(nk^2)$, чего достаточно для прохождения шестой подгруппы.

Для прохождения седьмой подгруппы применим несложную оптимизацию: не будем переходить из тех состояний динамики, которые невозможны. Пусть мы считаем значение динамики для вершины u и сейчас обновляем их для ее сына v . Тогда, перебирая, сколько путей мы возьмем из сына, будем перебирать это значение до величины $\min(k, size_v)$, где $size_v$ — размер поддерева вершины v . Также, когда мы перебираем, сколько путей мы берем из нашей текущей динамики dp' , будем перебирать это значение до величины $\min(k, size'_u)$, где $size'_u$ — суммарный размер всех обработанных детей u . Заметим, что ограничив перебор количества путей этими значениями, мы все еще учитываем все возможные переходы: действительно, нас не интересуют значения, большие k , так как всего мы хотим взять k путей, а также, нас не интересуют значения, большие размера соответствующих поддеревьев, так как из поддерева невозможно выбрать больше путей, чем в нем есть вершин.

Можно доказать, что благодаря этим оптимизациям асимптотика времени работы программы уменьшится до $O(nk)$, этого достаточно для получения 100 баллов.