

Разбор задачи «Упорядочивания»

В задаче нужно было посчитать количество топологических сортировок дерева, ребра которого ориентированы. Напомним, что топологической сортировкой графа называется такое упорядочивание вершин, при котором любое ребро ведет из вершины с меньшим номером в вершину с большим номером.

Решение на 16 баллов. Переберем все возможные перестановки чисел от 1 до n . Для каждой из них проверим, удовлетворяет ли она необходимым условиям. Итоговая сложность решения — $\mathcal{O}(n! \cdot n)$.

Решение на 32 балла. Воспользуемся динамическим программированием по подмножествам. Пусть $dp[S]$ — количество перестановок вершин из множества S , которые удовлетворяют необходимым условиям, где S — подмножество множества чисел от 1 до n . Чтобы пересчитывать динамику, переберем, какую вершину мы добавляем в множество, и если не существует ребер из новой вершины в текущее множество, ее можно добавить и обновить значение динамики.

Решения за полиномиальное время. Чтобы получить решение, работающее за полиномиальное время, подвесим дерево за произвольную вершину, то есть выберем одну вершину и сделаем ее корнем дерева, а для всех остальных вершин определим их родителя. Пусть $sz[u]$ — размер поддерева вершины u . Эту величину мы можем посчитать заранее обходом в глубину.

Решение для частного случая: все ребра вниз. Решим сначала более простой частный случай задачи: пусть все ребра направлены вниз, то есть от родителя к ребенку. В этом случае задачу можно решить с помощью динамического программирования по поддеревьям. Пусть $dp[u]$ — количество топологических сортировок поддерева вершины u . Пусть вершина u имеет k детей: v_1, v_2, \dots, v_k . Рассмотрим какую-нибудь топологическую сортировку поддерева u . На первом месте в ней всегда обязана стоять вершина u , так как все ребра в ее поддереве направлены вниз. После этого у нас останется $sz[u] - 1$ свободных позиций, которые займут вершины из поддерева u , кроме самой u . Заметим, что мы можем произвольным образом выбрать подмножество позиций, которые займут вершины из поддерева v_1 , затем произвольным образом выбрать подмножество позиций, которые займут вершины из поддерева v_2 , и так далее по всем сыновьям u . После этого мы можем каждое из поддеревьев вершин v_1, \dots, v_k топологически отсортировать. Несложно убедиться, что после этого мы всегда получим какую-то топологическую сортировку поддерева u , а также любая топологическая сортировка поддерева u может быть получена таким способом.

Итак, сначала нам нужно из $sz[u] - 1$ свободных позиций выбрать $sz[v_1]$ позиций для вершин поддерева v_1 . После этого нужно из $sz[u] - 1 - sz[v_1]$ свободных позиций выбрать $sz[v_2]$ позиций для вершин поддерева v_2 , и так далее. На i -м шаге у нас есть $sz[u] - 1 - sz[v_1] - \dots - sz[v_{i-1}]$ свободных позиций, из которых нужно выбрать $sz[v_i]$ для вершин поддерева v_i . Количество способов сделать это — это $C_{sz[u]-1-sz[v_1]-\dots-sz[v_{i-1}]}^{sz[v_i]}$. Здесь через C_n^k мы обозначаем количество сочетаний из n по k . Эти значения можно посчитать заранее с помощью треугольника Паскаля.

Таким образом, общее количество способов распределить места составляет

$$C_{sz[u]-1}^{sz[v_1]} \cdot C_{sz[u]-1-sz[v_1]}^{sz[v_2]} \cdot C_{sz[u]-1-sz[v_1]-sz[v_2]}^{sz[v_3]} \cdot \dots \cdot C_{sz[u]-1-sz[v_1]-\dots-sz[v_{k-1}]}^{sz[v_k]}$$

Полученный результат нужно также умножить на все значения $dp[v_1], dp[v_2], \dots, dp[v_k]$, так как после того, как мы распределили места, мы можем произвольным образом выбрать топологическую сортировку для каждого из поддеревьев v_1, \dots, v_k .

Посчитав таким образом динамику для всех поддеревьев исходного дерева, мы получим ответ для случая, когда все ребра ведут вниз.

Полное решение. Пусть теперь некоторые ребра ведут вверх. Удалим из дерева все ребра, ведущие вверх, и решим задачу для оставшихся ребер. Для этого внутри каждого получившегося дерева посчитаем динамику, описанную выше, а после этого скомбинируем ответы: чтобы получить топологическую сортировку всего графа, нужно произвольным образом выбрать для каждого из отдельных деревьев множество позиций, которые будут занимать вершины из этого дерева, а также топологическую сортировку внутри вершин этого дерева. Формула, по которой можно получить ответ, идентична той, с помощью которой мы считали динамику для первого случая.

Заметим, что не любое упорядочивание вершин среди тех, которые мы посчитали, подходит. Некоторые из полученных упорядочиваний нарушают условие для каких-то ребер, направленных

вверх. Пусть есть s ребер, направленных вверх. Пронумеруем их произвольным образом от 1 до s . Пусть A_i — множество упорядочиваний вершин, в которых конца ребра вверх под номером i расположены в неправильном порядке. Тогда, чтобы получить правильный ответ, нам нужно посчитать размер множества $A_1 \cup A_2 \cup \dots \cup A_s$ и вычесть его из ответа, который мы уже посчитали. Воспользуемся для этого формулой включений-исключений.

Пусть для всех подмножеств ребер, ведущих наверх, мы посчитали количество топологических сортировок, в которых для каждого из этих ребер (и, быть может, для каких-то других) условие нарушено. Тогда чтобы посчитать размер объединения множеств A_i , нужно сложить эти значения для подмножеств с нечетным числом ребер и вычесть значения для подмножеств с четным числом ребер.

Однако перебор всех возможных подмножеств ребер, ведущих наверх, занимает экспоненциальная время. Чтобы не делать этого, добавим несколько необходимых нам параметров динамики. Дополнительно нам понадобится знать, какова четность количества ребер вверх, для которых условие заведомо нарушено, а также размер текущего поддерева текущей вершины, если учитывать только ребра вниз. Иными словами, нам нужно хранить в состоянии динамики размер текущего поддерева вершины с учетом того, что некоторые ребра вверх мы удалили, а некоторые развернули в другую сторону. Пусть теперь $dp[u][size][parity]$ — это количество упорядочиваний поддерева вершины u , в которых текущее поддерево вершины u по ребрам вниз имеет размер $size$, а число ребер вверх, которые мы перевернули, дает остаток $parity$ от деления на 2. Будем пересчитывать эту динамику, добавляя детей очередной вершины по очереди. Каждый раз, когда мы встречаем ребро дерева, ведущее вниз, мы можем только добавить к текущему поддереву по ребрам вниз новую часть. Если же мы встречаем ребро вверх, то мы можем либо убрать его, либо развернуть, изменив при этом четность числа ребер, порядок на которых нарушен. Каждый раз, когда мы выкидываем очередное ребро вверх, будем домножать ответ на соответствующее число сочетаний. В конечном итоге в значениях динамики в корне дерева мы получим необходимые слагаемые для формулы включений-исключений.