Преступная сеть

Авторы задачи: Константин Бац и Даниил Орешников, разработчик: Даниил Орешников

Переформулируем задачу более простым образом. Дано подвешенное дерево, у каждой вершины есть вес, у каждого ребра есть длина. Требуется найти поддерево максимального веса, достижимое из своего корня за время T.

В первой подгруппе достаточно написать квадратичное решение — запустить обычный `dfs` из каждой вершины вниз по дереву и посчитать сумму весов вершин, достижимых за доступное время.

Во второй подгруппе гарантируется, что дерево является бамбуком, то есть у каждой вершины ровно один ребенок. В таком случае задача на дереве сводится к задаче на массиве — найти отрезок длины не более T с максимальной суммой. Такая задача решается с помощью префиксных сумм и метода двух указателей за линейное время.

Подгруппа с $T \leqslant 10$ и $t_i = 1$ решается подсчетом количества детей у каждой вершины на каждом уровне. Действительно, вершины на расстоянии t_* — это просто потомки вершины на t_* уровней дерева ниже. Находить количество детей на разных уровнях можно было с помощью эйлерова обхода или сливанием результатов, посчитанных для детей.

Следующие две подгруппы расчитаны на упрощенные вариации полного решения. Жюри же известно два полных решения данной задачи. Авторское решение заключается в следующем. Запустим один `dfs` и посчитаем для каждой вершины глубину h — расстояние от нее до корня дерева. Тогда вершина u и ее предок v находятся на расстоянии не более T тогда и только тогда, когда $h[u] - h[v] \leqslant T$.

Представим, что у нас есть декартово дерево всех детей вершины v с высотой вершин в качестве их ключей. Рассмотреть все вершины на расстоянии не более T от v — сделать `split` по ключу $h[v] + T$. Если поддерживать в декартовом дереве для каждой вершины сумму весов поддерева, то ответ на задачу при выборе стартовой вершины v получается как значение этой величины в левой части ДД после `split`.

Осталось понять как получать такое ДД. Будем идти от детей к родителям и сливать результаты для детей. При «перекладывании» элементов из меньшего дерева в большее каждый элемент будет переложен не более $\log n$ раз, асимптотика такого решения — $\mathcal{O}(n \log^2 n)$.

Альтернативное решение: можно заметить, что v попадет в ответ только для тех предков, для которых $h[u] \geqslant h[v] + T$. Множество таких u образует вертикальный путь в дереве, и вершину этого пути можно найти с помощью двоичных подъемов и бинпоиска. Таким образом, если вершина v попадает в ответ для всех вершин на пути $u_1 \to u_2$, то достаточно применить операцию $\texttt{answer}_u \leftarrow \texttt{answer}_u + (a_v + 1)$ для всех $u \in [u_1, u_2]$.

Операцию прибавления на вертикальном пути можно сделать отложенной, прибавив значение к u_2 и вычев его из p_{u_1}. После чего настоящие значения можно посчитать как сумму отложенных значений в поддереве. В конце надо выбрать вершину v с минимальным $\texttt{answer}_v$. Общее время работы такого решения равно $\mathcal{O}(n \log n)$.