

Цены на бензин

Для начала поймём, что от нас требуется. Дано дерево, в каждой вершине которого записан диапазон цен для этой вершины. Запрос это пара путей равной длины, цены на i -ых вершинах вдоль этих путей должны быть равны для всех i . Нужно найти количество способов расставить цены на вершинах для каждого префикса ограничений.

Начнём с медленного решения задачи. Будем хранить компоненты связности (в каждой цене на вершинах должны быть равны). Для каждой из них храним допустимый диапазон цен. Ответом будет произведение длин диапазонов по всем компонентам. Будем идти вдоль путей и объединять 2 вершины в одну компоненту с помощью СНМ. Это уже набирает 31 балл. Используем эвристику сжатия путей и ранговую эвристику в СНМ, также обходим пути запросов за линейное время от их длин - 47 баллов. Понятно, что для ускорения этого решения надо быстрее искать моменты, когда две вершины объединяются в одну компоненту.

Сначала разберём долгое решение. Сделаем heavy-light decomposition, с помощью которого будем хешировать пути в дереве, приняв за символ для вершины номер корня её компоненты. Теперь с помощью бинарного поиска будем искать первый момент, когда хеши на префиксах путей различаются, то есть две вершины объединяются в одну компоненту. С помощью переливаний обновим для вершин корни их компонент и дерево отрезков для hld. Получим n объединений, каждое найдём за $O(\log^2(n))$ с помощью hld. Также будет $O(n \cdot \log(n))$ обновлений в дереве отрезков из-за переливаний. На каждый запрос будет $O(\log^2(n))$ от hld. Итоговая асимптотика $O((n + q) \cdot \log^2(n))$.

Теперь приведём красивое решение данной задачи. Начнём с бамбука.

Заменим равенство цен на паре путей на две пары путей с длинами равными максимальной степени двойки, меньшей длины исходного пути (как в sparse table). Теперь длины путей всех ограничений стали равны степеням двойки. Будем перебирать степени двойки в порядке убывания 2^k , для каждого пути длины 2^k заведём вершину в графе, также заведём вершину для каждого такого пути в обратном порядке. Теперь ограничения задают рёбра в таком графе. Проведём их, выделим остовное дерево. Для каждого ребра из остова разделим ограничения на 2 ограничения с длинами путей в два раза меньше и продолжим процесс. На слое с длинами 1 получим нужное нам остовное дерево, которое будет отвечать за первые моменты, когда некоторые пары вершин объединялись в компоненты. Заметим, что с каждого слоя добавится вниз не более $2n$ рёбер, а также от запросов не более $2q$ рёбер. То есть каждый слой отработает за $O((n + q) \cdot \alpha(n))$, где $\alpha(n)$ - среднее время работы в снм, обратная функция Аккермана. Получили решение за $O((n + q) \cdot \alpha(n) \cdot \log(n))$.

Для полного решения на дереве для начала разобьём пару путей на три пары соответствующих вертикальных путей (возьмём из 4 конечных вершин этих путей самую близкую к lca пары вершин на её пути, тогда в пару к этому пути поставим вертикальный путь (часть другого пути), теперь получим один вертикальный путь и произвольный путь в дереве, разобьём второй путь по lca и первый по соответствующим длинам). Далее поступим аналогично бамбуку, только место вершины, отвечающей за отрезок, заведём вершину, отвечающую за бинарный подъём в дереве на высоту равную степени двойки. Итоговая асимптотика $O((n + q) \cdot \alpha(n) \cdot \log(n))$.