Homework 13

Dimitrov Blagoi

2019-05-26

Задание 1

Разобьём каждый запрос вида (k, x, l, r) на два: (k, l, r) и (x, l, r). Т.е. мы по отдельности будем прибавлять на отрезке число x и прибавлять арифм. посл. вида $0, k, 2 * k, ...$ Запросы прибавления числа на отрезке будем обрабатывать при помощи префикс сумм. Т.е. хранить массив p и делать p[l] += x и p[r + 1] -= x.

Прогрессии будем обрабатывать следующим образом: заведем массив q и будем хранить в q[i] все запросы, у которых l == i или r - 1 == i.

Пройдемся по массиву q, при этом будем поддерживать текущий шаг прогрессии - это сумма шагов всех сейчас рассматриваемых прогрессий (у которых l <= i <= r). Тогда к нашему элементу надо будет прибавить столько же, сколько мы прибавили к первому + этот шаг.

Всё. Такой алгоритм как раз работает за $O(n + m)$ т.к. нужно пройтись по массивам за $O(n)$ и посмотреть всего $O(m)$ событий.

■

Задание 2 и 3

Построим HLD. Теперь, на каждом пути в данном нам дереве построим Декартово Дерево по явному ключу. К роли ключа будет выступать высота вершины, ведь в любом пути в дереве не может быть двух вершин с одинаковой высотой.

Precalc:

Посчитаем высоты всех вершин + будем хранить информацию о том, какому пути какая вершина принадлежит + массив двочиных подъёмов + вершину с самой большой высотой на пути.

Query:

Когда мы помечаем вершину, то просто за $O(log\ n)$ удаляем ее из Декартовова Дерева соответствуюшего пути. Когда снимаем пометку - добавляем в дерево.

Отдельно рассмотрим поиск непомеченного LCA. Сначала найдем LCA за $O(log\ n)$. Это и есть наш ответ, если LCA - непомеченный. В противном случае, разобьем путь от LCA до корня на несколько, каждый из которых являтся какой-то частью какого-либо пути из декомпозиции. Рассмотрим все эти пути. Мы за $O(1)$ можем проверить, есть ли на пути из декомпозиции вершина, лежащая на нашем пути. Мы берем самую высокую вершину и смотри, находился ли она выше пересечения нашего пути и пути из декомпозиции. Если вершина не находиться там, где надо, переходим к следующему пути (тому, что выше). Как только вершина нашлась, надо найти самую низкую вершниу на префиксе этого пути до его пересечения с нашим. Это можем сделать за один запрос в Декартовом Дереве.

Таким образом мы обработаем не более $O(log\ n)$ путей, которые мы пропустим каждый за $O(1) + O(log\ n)$ на последний запрос. Таким образом получили нужную асимптотику.

■