Метро в Берляндии

Заметим, что после каждого запроса схема метро остается деревом. Тогда время, за которое можно пройти все метро и вернуться на начальную станцию составляет 2 * (количество ребер в дереве) минут. Но нам не нужно возвращаться на начальную станцию. В таком случае, если зафиксировать начальную и конечную станцию, то время за которое можно посетить все станции хотя-бы один раз составляет 2*(количество ребер в дереве) - (расстояние от начальной до конечной станции). Чтобы минимизировать эту величину, нужно выбрать начальную и конечную станцию так, чтобы расстояние между ними было максимальное. Такую величину величину еще называют диаметром дерева. Задача свелась к тому, что нам нужно уметь поддерживать количество ребер и диаметр в дереве после каждого запроса.

Научимся решать задачу, если у нас запросы только первого типа. Будем поддерживать концы диаметра дерева(пускай это будут вершина a и b). Представим, что мы добавили вершину c. Тогда нашим новым диаметром станет либо (a, b), либо (a, c), либо (b, c). Это следует из того, что если подвесить дерево за предок вершины c, то самой глубокой вершиной будет вершина a или b(Доказательство этого факта остается за читателями). И тогда из алгоритма поиска диаметра дерева через два BFS, получаем вышеупомянутый результат.

Теперь мы заметим, что каждая станция имеет отрезок, на котором это станция существует. Заведем дерево отрезков, а в нодах будем хранить список станций(если нода отвечает за отрезок $[l, r]$, то все станции обязаны существовать на этом отрезке). Сделаем обход по дереву обходу отрезков. Будем поддерживать количество ребер и текущий диаметр дерева. Перед тем, как зайти в ноду скопируем значения количества ребер и диаметра дерева из предка. Затем обновим эти параметры с учетом списка вершин. Как только мы окажемся в i листе дерева отрезков, мы получим ответ на i запрос.