Алгоритмы и структуры данных. Семинар 16. Деревья отрезков, LCA, разные задачи.

Григорьев Дмитрий БПМИ-163

Задача 4.

Заведем 3 указателя. Первый изначально в первой вернише, второй – во второй. Первый указатель поднимем на 1, второй на два, затем первый еще на три (чтобы расстояние до вершины отправления стало 4), затем второй на 6 (чтобы расстояние до вершины отправления стало 8), затем первый на 12 (расстояние станет 16) и так далее, на i-ом шагу мы двигаем (i mod 2 + 1) - ой указатель так, чтобы его расстояние до вершины отправления стало равно 2^i. Тогда указатели уже точно встретятся, когда они оба уже пересекут общего предка, и когда длина шага будет хотя бы 3 d, где d – разница в глубине (пусть длина шага равна 3 q, $q >= d$. Тогда один из указателей находится на расстоянии 2 q от своей вершины, а другой двигается от расстояния q до расстояния 4 q – если он выше, то он встретит второй указатель будучи на расстоянии 2 q - $d >= q$, иначе на расстоянии 2 $q + d <$ 4 q). Пройденное указателями расстояние пропорционально длине последнего совершенного шага, так что сложность O(ρ).

Когда указатели нашлись,считаем разницу растояний до вершин и поднимим нижнюю из них на один уровень. Дальше просто переходя к предкам найдем LCA.

Задача 8.

Эта структура данных – дерево отрезков снизу.

Задача 9.

Изначально будем считать, что количество различных цветов в поддереве – это размер поддерева.

Далее запомним вершины в порядке DFS, храня для каждой вершины предыдущюю вершиину этого же цвета.

Потом для текущей вершины и следующей такого же цвета, если таковая имеется, ищем LCA за $O(1)$ (алгоритм Фарах-Колтона и Бендера). И теперь обновляем количество различных цветов в поддереве: для LCA – вычитаем 1.

Теперь количество различных цветов в поддереве – сумма в поддереве, с учетом обновлений.

Время раюоты – $O(n)$, так как LCA мы ищем за $O(1)$ и сумма в поддереве – $O(n)$.