

## Условие

Есть дерево, каждое ребро может быть либо включено, либо выключено. Научитесь обрабатывать такие запросы за полилог:

- Изменить состояние всех ребер на пути от  $u$  до  $v$
- Найти число компонент связности по включенным ребрам

## Решение

Заметим, что в дереве число компонент связности = число выключенных ребер + 1. Тогда задача сводится к нахождению числа выключенных ребер.

Разобьем дерево по HLD и в каждом Heavy-сегменте будем поддерживать дерево отрезков, которое считает число выключенных ребер, т.е. выставленных битов в битовом отрезке. Мы умеем делать инвертирование битов на подотрезке, это уже было на практике. Light-ребра тоже помнят свою включенность.

Запрос на изменение состояния выглядит так:

1. Найдем LCA  $u, v$
2. На пути  $u \rightarrow \text{LCA}$  и  $v \rightarrow \text{LCA}$  найдем все Heavy-сегменты, через которые мы проходим. В этих сегментах найдем подотрезок, по которому мы проходим (*это делается за  $O(1)$ , если в каждом элементе Heavy хранить указатель на верхний элемент в Heavy*)
3. На каждом таком подотрезке инвертируем значения и из каждого Heavy передадим в родителя обновленное число выключенных ребер. Если Heavy-сегмент передал в Light-ребро значение, то это ребро передает вверх полученное +1, если Light выключено и просто полученное иначе.
4. Эти передачи доходят до корня, это значение и есть число компонент связности.