3 задача: LCA

4 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

Задано дерево с корнем, содержащее n ($1 \le n \le 100\,000$) вершин, пронумерованных от 0 до $n-1$. Требуется ответить на m ($1 \le m \le 10\,000\,000$) запросов о наименьшем общем предке для пары вершин. Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z. Числа $a_3, \ldots, a_{2m}$ генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i-1$-й запрос равен v, то i-й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$.
Для решения задачи можно использовать метод двоичного подъёма.

Формат входных данных.

- Первая строка содержит два числа: n и m. Корень дерева имеет номер 0.
- Вторая строка содержит $n - 1$ целых чисел, i-е из этих чисел равно номеру родителя вершины i.
- Третья строка содержит два целых числа в диапазоне от 0 до $n-1$: a_1 и a_2.
- Четвертая строка содержит три целых числа: x, y и z, эти числа неотрицательны и не превосходят 10^9.

Формат выходных данных.

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

in	out
3 2 0 1 2 1 1 1 0	2

2. Описание алгоритма

Пользуемся методом двоичного подъёма, описанного здесь:
https://neerc.ifmo.ru/wiki/index.php?title=Метод_двоичного_подъёма

3. Доказательство корректности

Следует из построения алгоритма

4. Время работы и дополнительная память

Препроцесинг:

$T = O(n \log n)$

Запрос:

$T = O(\log n)$

Память:

$M = O(n \log n)$

5. Доказательство времени работы

Всего состояний динамики $O(n \log n)$. Каждое состояние считается за $O(1)$. Поэтому суммарная сложность времени и памяти препроцессинга — $O(n \log n)$.

Весь алгоритм ответа на запрос состоит из изменения I от $L = \lceil log\ n \rceil$ до 0, а также проверки на каждом шаге за $O(1)$, является ли одна вершина предком другой. Следовательно, на каждый запрос будет найден ответ за $O\ (log\ n)$.