

# Мечтать не вредно

Авторы задачи: Константин Амеличев, Тимофей Федосеев

Разработчик задачи: Тимофей Федосеев

Сначала сделаем общее замечание, которое поможет нам в процессе решения всех подзадач. Рассмотрим порядок, в котором вершины будут удалены в исходном дереве, и дальше для простоты будем называть его порядком удаления.

Рассмотрим произвольную вершину  $v$  и обозначим за  $S_v$  множество вершин, идущих перед  $v$  в порядке удаления. Теперь определим оптимальную вершину для обнуления. Заметим, что бессмысленно обнулять вершину на пути от  $v$  до корня, так как это может только ухудшить позицию  $v$  в порядке удаления. При обнулении вершины  $s$ , не лежащей на пути до корня, позиция  $v$  в порядке удаления уменьшится на количество вершин из  $S_v$  в поддереве вершины  $s$ .

Таким образом, требуется найти поддерево, не содержащее вершину  $v$ , с наибольшим количеством вершин, идущих перед  $v$  в порядке удаления.

В дальнейшем под суммой в поддереве мы будем понимать количество вершин из  $S_v$  в этом поддереве.

- **Подзадача 1. Не более двух бамбуков (10 баллов)**

В этой подзадаче дерево состоит не более чем из двух бамбуков, подвешенных к корню. В этом случае найти порядок удаления можно методом двух указателей. Оптимальной вершиной для обнуления является корень одного из бамбуков. Для ответа на запросы достаточно пройти по порядку удаления и поддерживать сумму в обоих бамбуках.

- **Подзадача 2. Произвольное количество бамбуков (6 баллов)**

Решение аналогично подзадаче 1. Для нахождения порядка удаления нужно эффективно склеить несколько списков, это можно сделать с помощью структуры данных, например очереди с приоритетом или множества (set). Теперь при прохождении по порядку удаления будем поддерживать сумму в каждом бамбуке и дополнительно отслеживать два бамбука с максимальной суммой. Для ответа на запрос выберем лучший из них, который не содержит вершину запроса.

Теперь научимся проводить симуляцию процесса в общем случае. Будем поддерживать множество детей корня в структуре данных. На очередной итерации удалим из него вершину с максимальным значением и добавим её детей. В зависимости от выбранной структуры данных асимптотика составит  $O(n \log n)$  или  $O(n^2)$ . Для эффективной реализации подойдет очередь с приоритетом или множество (set).

- **Подзадача 3. Любая симуляция (8 баллов)**

В этой подзадаче требуется написать любую корректную симуляцию. Для ответа на запрос переберем вершину для обнуления и запустим симуляцию. Получим решение не хуже  $O(n^4)$ .

- **Подзадача 4.  $O(n^2 \log n)$  (13 баллов)**

Переберем вершину для обнуления, запустим симуляцию за  $O(n \log n)$  и обновим ответ для каждой вершины ее номером в получившемся порядке удаления.

- **Подзадача 5.  $O(n \log n + nq)$  (11 баллов)**

Будем идти по порядку удаления и поддерживать множество просмотренных вершин. Для ответа на запрос посчитаем суммы в поддеревьях и выберем максимальное, не содержащее вершину запроса.

- **Подзадача 6. Сбалансированное двоичное дерево (9 баллов)**

Будем идти по порядку удаления и поддерживать сумму в каждом поддереве. Эту информацию можно пересчитывать за  $O(\log n)$  при рассмотрении очередной вершины. Заметим, что оптимальная вершина для обнуления смежна с путём от вершины запроса до корня. Для ответа на запрос рассмотрим все такие вершины и получим решение за  $O(n \log n)$ .

- **Подзадача 7.  $O(n \log n + nh)$  (11 баллов)**

Будем действовать аналогично подзадаче 6. Помимо суммы в поддеревьях будем также для каждой вершины поддерживать двух детей с максимальной суммой. Пересчитывать такую информацию при добавлении вершины и находить поддерево, смежное с путём до корня и с максимальной суммой, можно за  $O(h)$ .

- **Подзадача 8. Куча на минимум (14 баллов)**

Можно заметить, что в этом случае каждое поддерево, смежное с путём от вершины запроса до корня, идёт в порядке удаления либо целиком до вершины запроса, либо целиком после.

Воспользуемся этим и предварительно посчитаем размеры поддеревьев, затем будем обходить дерево в глубину. Будем поддерживать размер оптимального поддерева, смежного с путём. Находясь в очередной вершине, отсортируем её детей по значению. Дети с большим значением будут целиком идти в порядке удаления перед детьми с меньшим значением. Запустим из них по очереди обход в глубину, обновив значение оптимального поддерева размером детей перед ними.

В результате такого обхода можно также выписать порядок удаления за линейное время. Получим решение за  $O(n)$ .

- **Подзадача 9. Полное решение  $O(n \log^2 n)$  (18 баллов)**

Аналогично подзадаче 6 будем идти по порядку удаления и поддерживать суммы в поддеревьях. Для этого воспользуемся структурой данных Heavy-Light Decomposition (HLD). При рассмотрении очередной вершины прибавим 1 на пути до корня. Для ответа на запрос вычтем  $\infty$  на пути до корня (чтобы не рассматривать поддеревья, содержащие вершину запроса), возьмем максимум в дереве и затем отменим вычитание.