

Задачи с лекции группы С на дерево отрезков

2 марта 2015 г.

Задача 1. Дана строка из нулей и единиц. Необходимо обрабатывать запросы присваивания на отрезке и запрос поиска позиции k -го нуля на отрезке.

Решение. Заведем дерево отрезков, в каждой вершине которого будем подсчитывать количество нулей и единиц. Запрос обновления выполняется довольно легко. Для нахождения k -го нуля на отрезке $[l; r]$ можно посчитать количество нулей на $[0; l - 1]$ (обозначим его x) и найти $(k + x)$ -й ноль на префиксе (можно искать и напрямую на отрезке, но задача на префиксе всегда проще). Для нахождения k -го нуля на префиксе можно использовать спуск по дереву: если в левом сыне корня есть $t \geq k$ нулей, то рекурсивно перейдем налево, иначе нужно перейти направо и найти там $(k - t)$ -й ноль.

Задача 2. Дан массив чисел. Необходимо обрабатывать запросы изменения в точке, а также запросы «найти подотрезок отрезка $[l; r]$ с максимальной суммой»

Решение. Заведем дерево отрезков, в каждой вершине которого будем поддерживать следующую информацию

- сумма на отрезке
- максимальная сумма на подотрезке данного отрезка
- максимальная сумма на префиксе
- максимальная сумма на суффиксе

Последние две величины важны для пересчета второй при слиянии информации из сыновей в предка. Продумывание процедуры слияния оставим как простое упражнение. Умея объединять информацию соседних на отрезках, мы получаем обновление и запрос за $O(\log n)$.

Упражнение. Решить ту же задачу, но с запросом присваивания на отрезке

Задача 3. Дан массив цифр от 0 до 9. Необходимо обрабатывать запросы «увеличить цифру на 1» на отрезке (при этом 9 переходит в 0), а также запросы на сумму чисел на отрезке.

Решение. Теперь в дереве отрезков мы будем хранить целый массив чисел a_i : сколько цифр каждого типа есть на отрезке. Ясно, что эту информацию легко объединять одним проходом по массиву. Чтобы посчитать сумму, нам нужно просто найти $\sum_{i=0}^9 a_i \cdot i$. Итак, мы получаем ответ на запрос за время $O(10 \cdot \log n)$.

Задача 4. В парке аттракционов есть американские горки длиной в n рельсов ($n \leq 10^9$). У каждой секции рельс есть наклон — положительное или отрицательное число, выражающее, насколько конец рельса выше или ниже конца. Изначально все наклоны равны 0. Поступают

запросы «установить наклон всех рельс на $[l; r]$ равным d » и «запустить с начала трека тележку с энергией, достаточной, чтобы подняться на y единиц высоты вверх и посчитать, сколько рельс она преодолеет». То есть тележка останавливается, как только достигнет высоты больше y .

Решение. Представим пока, что $n \leq 10^5$. Тогда задача решается деревом отрезков со следующей информацией в вершине

- суммарный наклон всех рельс на отрезке
- максимальная высота на отрезке относительно его начала (не меньше нуля)

Эту информацию легко объединять. Также нетрудно сделать и присваивание на отрезке. Осталось научиться только обрабатывать запрос. Здесь нам надо снова сделать спуск по дереву: нужно проверить, преодолеет ли тележка рельсы из левого поддерева, если да, то перейти вправо, запомнив, что тележка потратила (или преобрела) сколько-то энергии на подотрезке слева. Иначе нужно просто перейти влево и искать ответ там.

Теперь, когда $n \leq 10^9$ нам осталось только воспользоваться техникой «ленивого» дерева отрезков: написав дерево отрезков на указателях, мы будем создавать вершины только когда нам нужно посетить их в запросе. Так, мы достигнем асимптотики $O(\log n)$ времени и памяти на запрос.

Замечание. Задача предлагалась на IOI 2005, желающие могут попробовать сдать ее здесь.

Задача 5. Дан массив чисел. Необходимо обрабатывать запросы изменения отдельного числа и запросы «сколько есть чисел на $[l; r]$, таких, что их значение лежит от a до b .»

Решение. Если бы изменений не было, мы бы могли воспользоваться техникой сохранения отсортированного подмассива в каждой вершине дерева отрезков. Однако, перестраивать массив при обновлении долго, поэтому нужно сделать иначе. Возьмем вместо массива любую структуру, которая умеет делать следующие операции

- Добавить в себя число
- Удалить число
- Посчитать, сколько чисел в ней находятся от a до b по значению

В качестве такой структуры можно выбрать, например, декартово дерево (или красно-черное дерево с порядковой статистикой, доступное в недрах некоторых компиляторов. Также, если запрос был бы не посчитать количество чисел от a до b , а, например, посчитать число вхождений числа x , то можно использовать `std::map`).

Тогда для запроса нужно просто разбить отрезок деревом отрезков на $O(\log n)$ кусочков и просуммировать запросы к структурам на этих кусочках.

Для обновления нам нужно подняться вверх от одной вершины и во всех структурах на пути выкинуть «старое» число, добавив «новое».