МФТИ, ФПМИ

Алгоритмы и структуры данных, осень 2021

Семинар №8. Дерево отрезков (1)

1. Дан массив чисел $a_1, \ldots, a_n$. Обработайте q запросов двух видов: а) увеличить все числа на отрезке $[l, r]$ на величину val; б) сообщить число на позиции pos. Асимптотика: $O(n + q \log n)$. Обойдитесь без проталкиваний.

2. К массиву $a_1, a_2, \ldots, a_n$ поступают запросы изменения элемента в точке. После каждого запроса сообщите подотрезок массива с максимальной суммой (он может иметь нулевую длину). Асимптотика: $O(n + q \log n)$, где q – число запросов.

3. Дан массив длины n из нулей и единиц. Обработайте два типа запросов: а) обновить элемент ($a_{pos} := val$); б) вывести позицию k-го нуля на отрезке $[l, r]$. Ответ на запрос за $O(\log n)$. Не сводите задачу к отрезку $[1, r]$.

4. Дан массив чисел $a_1, \ldots, a_n$. Обработайте q запросов двух видов: а) выполнить присваивание в точке (то есть $a_{pos} := x$); б) по позиции pos и числу val сообщить длину максимального по длине отрезка массива, содержащего pos, все элементы которого не превосходят val. Асимптотика: $O(n + q \log n)$.

5. К массиву чисел длины n поступают запросы трёх типов: а) выполнить присваивание на отрезке (то есть $a_l := val, \ldots, a_r := val$); б) выполнить увеличение на отрезке (то есть a_l увеличить на $x, \ldots, a_r$ увеличить на x); в) сообщить сумму на подотрезке. Ответ на запрос за $O(\log n)$.

6. Дана строка s из круглых открывающих и закрывающих скобок длины n. Обработайте q запросов двух видов: а) изменить символ на позиции pos (то есть заменить тип скобки s_{pos}); б) сообщить, является ли подстрока на позициях с l по r правильной скобочной последовательностью. Обработайте все запросы за $O(n + q \log n)$.

7. Вспомним алгоритм с лекции, как находить количество чисел $\geqslant x$ на отрезке статического массива. Для этого строим дерево `MergeSort`, затем нужный отрезок $[l, r]$ разбиваем на $O(\log n)$ отрезков в этом дереве, наконец, в каждом из них запускаем бинарный поиск. Приведите пример запроса, который требует $\Omega(\log^2 n)$ действий при такой обработке.

8. На числовой прямой расположено n отрезков, i-й из которых начинается в точке l_i, заканчивается в точке r_i и имеет вес w_i. Известно, что объединение всех этих отрезков составляет отрезок $[1, m]$. За $O((n + m) \log(n + m))$ найдите такое подсемейство отрезков, что они покрывают весь $[1, m]$, и при этом разность между максимальным и минимальным весом отрезка подсемейства минимальна.

1. В каждой вершине дерева храните число, отвечающее величине, которую нужно прибавить ко всем числам поддерева. Проталкивать ничего не нужно. При запросе в точке нужно пройтись по пути от корня и сложить все эти обновления.
2. Можно обобщить эту задачу и отвечать на запрос "по индексам l и r сообщить максимальную сумму среди подотрезков отрезка $[l, r]$ массива". Для этого в каждой вершине дерева отрезков нужно хранить собственно величину ответа. Как склеивать результаты из двух вершин? Нужно поддерживать вспомогательные поля — максимальная сумма среди всех префиксов и среди всех суффиксов. Какое четвёртое поле также нужно хранить для пересчёта?
3. В вершине храните количество нулей в поддереве. Удобно представить запрос в виде $O(\log n)$ вершин в дереве отрезков. Далее нужно сделать спуск в одном из этих поддеревьев.
4. Храните максимум в поддереве, воспользуйтесь идеей спуска.
5. Нужно договориться с собой, что сначала выполняется присваивание, а потом прибавление.
6. Вспомните критерий правильности через баланс. Поймите, как меняется баланс при изменении одного символа и как проверять корректность подстроки.
7. Подойдёт такой отрезок, который в дереве отрезков покрывается вершинами размеров $1, 2, 4, 8, \ldots, 2^k$. Сумма логарифмов этих размеров есть $\Omega(k^2)$.
8. Отсортируйте все отрезки в порядке возрастания стоимостей. Для каждого отрезка в этом порядке найдите ближайший справа такой, что объединение всех отрезков между ними образует $[1, m]$. Воспользуйтесь идеей двух указателей.