Домашнее задание по АиСД №1

Эмиль Гарипов M3138

2020-04-19

Задача №1

Значение в вершине

В вершине будем хранить обычную сумму(назовем ее sum) на отрезке, а так же ту сумму, которую нас просят посчитать(назовем ее sum'):

$$\sum_{i=L}^{R} a_i \cdot (i - L + 1) =: sum'$$

.

Ответы на запросы

Построим дерево отрезков на массиве суффиксных сумм.

I **Изменение элемента**

При изменении элемента массива нужно просто пересчитать значения sum и sum'.

$$sum_v = sum_{2\cdot v+1} + sum_{2\cdot v+2}$$

$$sum'_v = sum'_{2\cdot v+1} + sum'_{2\cdot v+2} + sum_{2\cdot v+2} \cdot (m - l - 1)$$

II Запрос суммы sum'

Как в обычном дереве отрезков ответим на запрос суммы, только в вершинах будем брать значения sum'.

Время работы

Итого, ответ на запросы за $\mathcal{O}(\log_2 n)$.

Задача №2

Решение

Решим задачу с помощью дерева отрезков, сканирующей прямой и сжатия координат.

Для каждого прямоугольника(пусть их n штук, а так же будем считать, что прямугольник задается левым нижним и правым верхним углами) запишем y-координаты каждого из углов в массив, затем отсортируем этот массив. Пройдемся по этому массиву и каждой координате в соответствие поставим ее индекс в этом массиве используя хэш-талбицу с так называемым «идеальным хэшированием» (так как наши данные статичны, а добавление пары (Значение; Ключ) и ответ на запрос значения по ключу выполняются за $\mathcal{O}(1)$).

Итак, мы «сжали» координаты, на координатах сохранился порядок и $\max\limits_{t'} t' = \mathcal{O}(n)$, где t пробегает все значения новых(«сжатых») координат.

Последнее нам понадобится для того, чтобы построить дерево отрезков по y-координатам с временем ответа на запросы максимума на отрезке и прибавления на отрезке $\mathcal{O}(\log_2 n)$.

Будем поддерживать для y-координат дерево отрезков на максимум, для каждой y-координаты дерево хранит количество прямоугольников, которые покрывают ее(т. е. листьям соответствуют y-координаты, каждая вершина хранит максимум на отрезке, за который она отвечает, и индекс, где этот максимум достигается). Изначально все значения в дереве отрезков нули.

Теперь сканирующей прямой по x-координатам пройдемся по углам прямоугольников в порядке возрастания их x-координаты.

Введем обозначения:

- $y1$ — нижняя y-координата прямоугольника с текущим углом,
- $y2$ — соответственно верхняя.

Далее есть несколько случаев:

- Если очередной угол — левый нижний: прибавим в дереве отрезков 1 на отрезке $[y1,\ y2]$.
- Если очередной угол — правый верхний: запросим максимум на отрезке $[l,\ r]$ и запомним его. Если этот максимум превышает значение уже посчитанного нами ответа, то обновляем ответ, запоминаем точку(ее x-координата равна x-координате текущего угла, а y-координату мы получим по запросу в дереве отрезков), в которой достигся этот максимум.

После прохождения сканирующей прямой по всем x-координатам углов ответ будет посчитан. Но необходимо найти значение y-координаты полученной точки из входных данных, это сделаем просто проходом по массиву прямоугольников за $\mathcal{O}(n)$. Будем брать очередную y координату из входных данных и смотреть на ее значение в хэш-таблице. Если оно совпадает с тем значением y-координаты, что у нас записано, то запоминаем его как y-координату ответа и прекращаем поиск. Таким образом мы восстановим значение y-координаты искомой точки.

Время работы

Оценим время: сначала делается сортировка($\mathcal{O}(n \cdot \log_2 n)$), потом обработка прямоугольников (всего их n штук, каждый угол прямоугольника обрабатывается запросом в дереве отрезков $\mathcal{O}(\log_2 n)$, так же получаем $\mathcal{O}(n \cdot \log_2 n)$ на все прямоугольники).

Итого: $\mathcal{O}(n \cdot \log_2 n)$.

Подробнее об операциях в дереве отрезков

Сведем прибавление на отрезке к прибавлению в точке(как делали это в задаче на практике). А именно, сформируем новый массив a', значения которого считаются следующим образом:

$$a'_0 = a_0, a'_1 = a_1 - a_0, a'_2 = a_2 - a_1, \ldots, a'_i = a_i - a_{i-1}$$

Заметим, что значение элемента a_i равно сумме на префиксе длины $(i+1)$ в массиве a'. Тогда построим дерево отрезков на массиве a'

I **Прибавление на отрезке**

Запрос прибавления на отрезке $[l, r]$ значения x в массиве a сводится к прибавлению в точке следующим образом:

- Нужно прибавить в массиве a' к l-ому элементу x и вычесть из $(r+1)$-го элемента x.

Так как для вычисления значения элемента, который стоял в массиве a на i-ой позиции надо посчитать сумму на префиксе длины i в массиве a', после описанных операций все элементы в массиве a' до l-го и после r-го не поменяются, а на нужном нам отрезке ко всем элементам прибавиться x.

Изменение элемента в дереве отрезков выполняется за $\mathcal{O}(\log_2(n))$, соответственно прибавление на отрезке мы выполняем за такое же время.

II **Максимум на отрезке** $[l,\ r]$

Для вычисления максимума(назовем это значение M) на отрезке надо взять сумму максимального префикса, который заканчивается между l и r. Для этого будем спускаться по дереву и покрывать наш отрезок из запроса отрезками, за которые отвечает вершины дерева отрезков и вычислять на этих отрезках максимум(в вершинах будут хранится актуальные значения, так как при изменении мы будем корретно обновлять информацию). Теперь, когда мы знаем M, чтобы получить искомое значение, надо к M прибавить sum_l, где sum_l — сумма на отрезке $[0,\ l-1]$.

В каждой вершине будем хранить максимальную сумму префикса отрезка(назовен это значение $maxp$), за который отвечает вершина, и сумму на всем отрезке(назовем ее $sumsegm_v$). При изменении элемента(элементы меняются при прибавлении на отрезке) будем пересчитывать сумму на отрезке, за который отвечает вершина и вычислять $maxp_v$(v — вершина) по формуле

$$maxp_v = \max(maxp_{2\cdot v+1}, sumsegm_{2\cdot v_1} + maxp_{2\cdot v+2})$$

Итого, ответ на запрос работает за $\mathcal{O}(\log_2(n))$, так как мы просто спускаемся по дереву, высота которого $\mathcal{O}(\log_2(n))$ и при изменениях пересчитываем значения в вершинах.

Задача №4

Что от нас хотят?

Докажем, что значение, которое нужно посчитать по запросу — это просто максимальная сумма на суффиксе:

- Если посмотреть, как переменния s меняется с увеличением индекса i, можно заматетить, что первое значение которое примет s будет 0, затем будет какая-то последовательность положительных значений, затем снова 0, снова положительные значения, ноль и т. д. То есть нам требуется посчитать значение s после последнего обнуления. Покажем, что это значение есть ни что иное как максимальная сумма на суффиксе.

  1. **Докажем, что слева от самого правого максимального суффикса стоит 0.**
     Рассмотрим суффикс исходного массива, с максимальной суммой, а среди всех таких, самый правый. Слева от этого суффикса всегда стоит ноль, так как если бы там был не ноль, то мы бы могли добавить элемент слева к нашему суффиксу и он бы только увеличился, что противоречит максимальности нашего суффика.
  2. **Докажем, что справа от самого правого суффикса с максимальной суммой, не может стоят 0**.
     Если бы справа от максимального суффикса стоял 0, тогда мы могли сдвинуть границу суффикса правее, что противоречит тому, что мы выбираем самый правый суффикс.

Вывод: значение, которое требуется посчитать — самый правый суффикс с максимальной суммой.

Ответы на запросы

Построим дерево отрезков на массиве суффиксных сумм.

I **Прибавление на отрезке**

Изменение элемента сводится к прибавлению на отрезке, нужно просто прибавить всем суффиксам, в которые входит этот элемент, разность между старым значением и новым(все эти суффиксы лежат на префиксе массива суффиксных сумм). А это в свою очередь сводится к изменению в точке(подробно описано в решении 2-ой задачи).

II Поиск суффикса с макисмальной суммой

Поиск максимального суффикса сводится к максимуму на всем массиве(см. решение 2-о й задачи).

Время работы

Итого, ответ на запросы за $\mathcal{O}(\log_2 n)$.