Алгоритмы и Структуры Данных ДЗ-3

Гарипов Роман М3138

08.06.2020

Задача №1

Для начала научимся прибавлять константу на всём отрезке

Такая задача была в лабораторной. Будем работать с массивом `d_a[i] = a[i] - a[i - 1]`. Чтобы прибавить на отрезке $[l; r]$ значение x надо сделать два изменения массива :

- `d_a[l] += x`
- `d_a[r + 1] -= x`

Обработав все запросы, можно вычислить исходный массив с выполнеными операциями на отрезках так : посчитать префиксную сумму на массиве `d_a`.

Переходим к прибавлению прогрессии на отрезке

Для арифметической прогрессии характерно то, что разность соседних элементов это какая-то константа `k`, поэтому логично будет завести массив разностей массива разностей, так называемый массив `d_d_a[i]`. Тогда запросу прибавления прогрессии `(k, x, l, r)` будут соотвествовать следующие изменения массива :

- `d_d_a[l] += x`;
  `d_d_a[l + 1] += x`;
- `d_d_a[l + 1] += k`;
  `d_d_a[r + 1] -= k`;
- `d_d_a[r + 1] -= (x + k` $\cdot$ `(len - 1))`;
  `d_d_a[r + 2] += (x + k` $\cdot$ `(len - 1))`;

Мы явно сделали изменения в массиве в тех местах, в которых разность разностей соседних элементов поменялась. Чтобы найти итоговую последовательность, надо дважды посчитать массив префиксных сумм массива `d_d_a[i]`, потом посчитать префикс суммы на этом массиве префикс сумм, т е проще говоря дважды посчитать префикс суммы. Так как по условию задачи массив изначально проинициализован числами `a[i]`, нужно сложить соотвествущие элементы двух массивов чтобы получить ответ на задачу.

Задача №4

Посчитаем двоичные подъемы, чтобы уметь искать `lca` двух вершин. Так же построим `heavy-light декомпозицию`, в которой для каждого пути будем хранить дерево отрезков, в котором будем хранить самую левую и самую правую непомеченную вершину на нём(не думаю что стоит много говорить о том как это делать : искать минимум и его позицию на отрезке, массив будет состоять из 0 и 1 - непомеченные и помеченные вершины).

Запрос - пометить вершину `v`

- Возьмём путь на котором лежит вершина `v`.

  Сделаем запрос присвоения 1 в точке соответствующей вершине `v` в ДО на этом пути.

Запрос - найти самого нижнего непомеченного общего предка для `u` и `v`

- Найдём `lca(u, v)` $=$ `x` используя двоичные подъёмы.
- Теперь будем подниматься по путям нашей декомпозиции от вершины `x`.

  Поднимаясь от x до корня нашего дерева, мы обеспечим то, что ответ на запрос будет какой-то общий предок вершин `u` и `v`.
- Находясь на очередном пути, надо понять есть ли на нём хотя бы одна непомеченная вершина. Возьмем позицию минимума на этом пути в корне дерева отрезков, если эта вершина выше чем вершина `x` то на этом пути есть хотя бы одна непомеченная вершина которая является общим предком вершин `v` и `u`. Будем подниматься по декомпозиции до тех пор, пока не найдем такой путь.
- Когда нашли такой путь, запросим у дерева отрезков самый правый минимум на префиксе нашего пути от головы пути до вершины на которой сейчас стоим.

Итого : Предпосчёт за $\mathcal{O}(n\log(n))$, ответ на запрос за $\mathcal{O}(\log(n))$, т.к первый запрос это просто присвоение в дереве отрезков, второй запрос - подняться по декомпозиции - не более $\mathcal{O}(\log(n))$ $+$ один запрос в дереве отрезков - $\mathcal{O}(\log(n))$

Можно избавиться от двоичных подъёмов и искать `lca` подъёмом по декомпозии, это так же будет работать за $\mathcal{O}(\log(n))$, но предпосчёт будет занимать $\mathcal{O}(n)$ времени.

Задача №5

Аналогично задаче №4, для запроса снятия отметки с вершины надо будет присваивать 0 в точке.