

Семинар к лекции 9.

#вшпи #аисд #структуры_данных

Автор конспекта: Гридин Михаил

Этот конспект является логическим продолжением лекции [Лекция 9. Дерево Фенвика.](#)

Задача 1

Что умеет дерево Фенвика?

Операция	Может ли?
(+)	да
(·)	необратима относительно 0. Непонятно, см. далее
НОД()	нет, необратим
НОК()	нет, необратим
	нет
&	нет
(⊕)	да

Задача 2

Fenwick, операция (·):

- $Update(\cdot)$
- $GetProduct([l, r])$

Идея: будем хранить дерево Фенвика на количество нулей на отрезке и обновление в точке.

Теперь, чтобы ответить на запрос $GetProduct$ будем считать число 0 нейтральным элементом и будем считать, что $a \cdot 0 = 0 \cdot a = a$. Тогда если $GetZeros([l, r]) > 0$, то ответ 0, иначе $\frac{GetPrefix([1, r])}{GetPrefix([1, l])}$ и считаем обратное по модулю.

Задача 3

Fenwick, операция $getMax$:

- $Update(delta)$

- $getMax([l, r])$
не умеем, поскольку max - не обратимая операция.
Попробуем за $O(\log^2 n)$.

Нужно ограничение $\delta \geq 0$ или искать max только на префиксе $[1, r]$. Тогда точно за $O(\log n)$ получается. Если же $\delta < 0$, то мы можем пересчитать максимум на отрезке длины степени двойки за $O(\log n)$ через остальные отрезки. Итого асимптотика $O(\log^2 n)$. На произвольном отрезке $[l, r]$ искать максимум в дереве Фенвика пока не умеем.

Задача 4

Fenwick, операция +:

- $Add(\delta, [l, r])$
- $GetSum([l, r]) = PrefixSum(r) - PrefixSum(l - 1)$

сначала научимся делать $Get(i)$ в точке. Тогда в разностном массиве ($diff[i] = a[i] - a[i - 1]$) на отрезке будет $add(l, \delta), add(r + 1, -\delta)$. а значение в точке $get(i) = getPrefix_{diff}(i)$.

Теперь $GetSum([l, r])$. Делать $push$ сверху вниз не получится, потому что пушить вниз за быстро в дереве Фенвика не умеем.

Рассмотрим массив $diff$:

$$diff[i] = a[i] - a[i - 1], \quad a[0] = 0$$

$$a[i] = \sum_{j=1}^i diff[j] \implies$$

$$PrefixSum(r) = \sum_{i=1}^r a_i = \sum_{i=1}^r \cdot \underbrace{\left(\sum_{j=1}^i diff[j] \right)}_{\text{индекс 1 встречается } n \text{ раз, индекс 2 встречается } n-1 \text{ и т.д.}} =$$

$$= \sum_{i=1}^r (r - i + 1)diff[i] = \sum_{i=1}^r \left(\underbrace{r \cdot diff[i]}_{\text{умеем считать}} - (i - 1) \cdot diff[i] \right) =$$

$$= r \sum_{i=1}^r diff[i] - \underbrace{\left(\sum_{i=1}^r ((i - 1) \cdot diff[i]) \right)}_{\text{построим второго Фенвика на таком массиве}}$$

Заметим, что при обновлении на отрезке что в первом, что во втором Фенвике нужно обновить ровно 2 элемента, так что мы можем считать всё за $O(\log n)$. Задача решена.

Задача 5.

Fenwick 3D, операция (+):

- $Update(x, y, z, \Delta)$
- $GetSum(P_0, P_1)$, где $P = (x, y, z)$.

Хотим выразить через сумму на префиксе $GetPrefix(x, y, z)$.

По формуле включений-исключений:

$$\begin{aligned} GetSum(x_1, x_2, y_1, y_2, z_1, z_2) &= ps(x_2, y_2, z_2) - \\ &- (ps(x_1, y_2, z_2) + ps(x_2, y_1, z_2) + ps(x_2, y_2, z_1)) + \\ &+ (ps(x_2, y_1, z_1) + ps(x_1, y_2, z_1) + ps(x_1, y_1, z_2)) - \\ &- ps(x_1, y_1, z_1) \end{aligned}$$

слагаемые, у которых нечётное количество переменных с индексом 2 берём с плюсом, другие с минусом. Всегда получается 2^d слагаемых, где d - размерность.

Задача 6 (встречный Фенвик)

Fenwick*:

- $Update(i, \Delta), \Delta \geq 0$
- $GetMax([l, r])$

Сделаем "развёрнутый Фенвик" наоборот. Пусть к нам приходит запрос на $[l, r]$. Сделаем два подъёма по деревьям Фенвика. В обычном дереве будем подниматься от r к верху, пока отрезки покрывают элементы правее l , аналогично поднимемся во втором дереве Фенвика от l к верху, пока отрезки покрывают элементы левее r . Возьмём максимум из всех значений. Это будет ответом.

Теорема. При описанном алгоритме каждый элемент любого исходного отрезка $[l, r]$ покрывается хотя бы одним из рассмотренных отрезков в каком-то из деревьев.

□ Представим встречное дерево Фенвика 2^n на 2^n и посмотрим на него, как на дерево отрезков ([Лекция 8. Разреженная таблица. Дерево отрезков.](#)). В нём существует отрезок $[1 \dots 2^n]$. Оставшуюся часть можно разбить на 2 поддерева, т.е. отрезок $[1 \dots n]$ разбивается на подотрезки $[1 \dots \frac{n}{2}], [\frac{n}{2} + 1 \dots n]$. В итоге получается структура обычного дерева отрезков, для которого известно указанное выше утверждение. ■