

Если явно не сказано иное, каждую задачу требуется решить за $O(n \log^k n)$.

1 Запрос с зависимостью от внешнего значения

Задача (Заочный этап Открытой олимпиады школьников 2015 года). Дана строка из десятичных цифр. Требуется производить два вида операций:

- Поменять значение в точке.
- Посчитать сумму чисел, записями которых являются все подстроки данного отрезка, с учетом кратности. Числа, запись которых начинается на 0, считаются не должными.

Решение. Без условия на исключение чисел, запись которых начинается с нуля, задача решается деревом отрезков, в вершине которого хранится значение $a_l \cdot 10^0 + a_{l+1} \cdot 10^1 + \dots + a_r \cdot 10^{r-l+1}$ и что-то подобное (насколько вообще подробно надо выписывать решение?). Для того, чтобы исключать из суммы невалидные числа, требуется при запросе в вершину дерева отрезков передавать количество нулей в массиве слева от отрезка, соответствующего этой вершине, и пересчитывать это значение во время спуска по дереву отрезков.

2 Несколько деревьев отрезков

Задача (Загадочное устройство, Северный четвертьфинал ACM). Дан массив из чисел, не превосходящих 2010. Требуется производить два вида операций:

- Вывести сумму на данном отрезке.
- Заменить каждое число на отрезке на его квадрат по модулю 2010.

Решение. Заметим, что для любого $0 \leq a < 2010$ последовательность a, a^2, a^4, \dots периодична с периодом 10 и, возможно, некоторым небольшим предпериодом.

Забудем про предпериод. Заведем 10 деревьев отрезков. В i -м дереве в позиции j будет храниться значение $a_j^{2^i}$. Теперь при возведении в квадрат отрезка найдем, какие вершины соответствуют ему в дереве отрезков, и циклически сдвинем эти вершины во всех 10 деревьях.

Для обработки предпериода можно завести еще одно дерево отрезков, значением в котором будет максимальное расстояние до периода среди всех чисел на отрезке. При обработке запроса будем обходить его в глубину, пропуская вершины, в которых все числа достигли периода. Суммарно этот обход также работает за $O(n \log n)$.

Задача. Заданы два числа A, B . Требуется производить два вида операций:

- Прибавить значение в позициях $x, x + A, \dots, y - A, y$
- Посчитать сумму в позициях $x, x + B, \dots, y - B, y$

Решение. Корневая декомпозиция.

- $A > \sqrt{n}, B > \sqrt{n}$. Решаем наивно: каждый запрос затрагивает не более \sqrt{n} позиций
- $A < \sqrt{n}, B > \sqrt{n}$. Заведем A деревьев отрезков для каждого остатка по модулю A . Первый запрос — прибавление на отрезке в одном из деревьев. Второй — не более \sqrt{n} запросов в точке к какому-то из деревьев.
- $A > \sqrt{n}, B < \sqrt{n}$. Аналогично предыдущему.
- $A < \sqrt{n}, B < \sqrt{n}$. Заведем $\text{lcm}(A, B)$ деревьев отрезков. Первый запрос — прибавление на отрезке не более B раз. Второй — сумма на отрезке не более A раз.

3 Использование структуры дерева отрезков в других задачах

Задача (SNWS 2015). На прямой расположены бомбы. Каждая бомба задается координатой и радиусом действия. При подрыве бомбы взрываются все бомбы в радиусе действия. Требуется для каждой бомбы определить, сколько бомб взорвется при ее прямом подрыве.

Решение. Заметим, что бомбы, взрывающиеся при подрыве данной, образуют отрезок. Проведем ребро между двумя бомбами, если вторая подрывается при взрыве первой, и построим конденсацию этого графа. После этого на ациклическом графе считаем динамику «самая левая и самая правая бомба, которая взорвется при подрыве этой компоненты связности».

Как строить конденсацию за $O(n \log n)$? Построим фиктивное дерево отрезков на последовательности бомб. Вершины дерева отрезков также будут вершинами графа, который мы конденсируем. Проведем ребра между вершиной дерева отрезков и ее сыновьями. Пусть бомба x покрывает отрезок бомб $[l, r]$. Вместо $r - l + 1$ ребер добавим $O(\log n)$ в те вершины дерева отрезков, которые соответствуют отрезку $[l, r]$. Можно заметить, что конденсация этого графа совпадает с требуемой конденсацией, если не учитывать добавленные фиктивные вершины.

Задача (Dynamic Connectivity Offline). В граф добавляются и удаляются ребра. Требуется после каждого изменения печатать количество компонент связности графа.

Решение. Построим дерево отрезков на временах запросов. Для каждого ребра определим интервалы, в течение которых ребро находится в графе. Для каждого такого отрезка добавим номер этого ребра во все соответствующие вершины дерева отрезков (в вершине хранится список ребер).

Будем обходить дерево отрезков обходом в глубину, поддерживая систему непересекающихся множеств с ребрами, которые встретились на пути. При приходе в очередную вершину добавляем новые ребра в СНМ, при возврате из рекурсии откатываем СНМ в предыдущее состояние. Для этого нужно отказаться от эвристики сжатия путей и использовать только ранговую эвристику.