22) Дерево отрезков

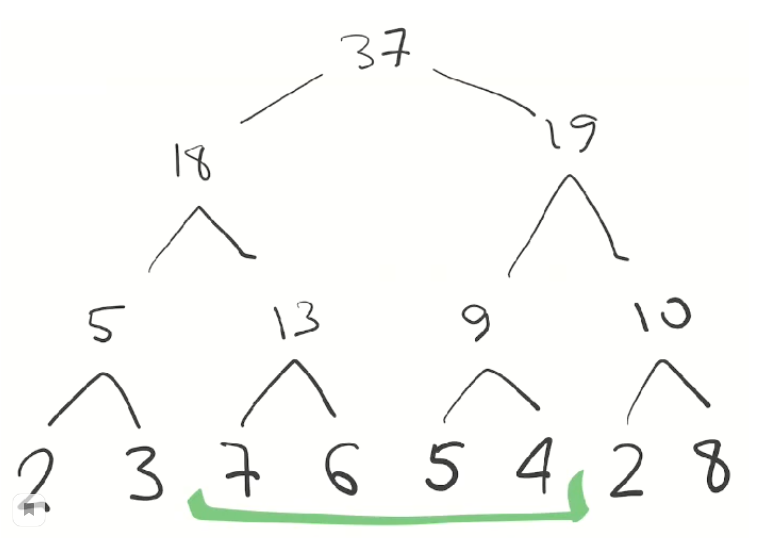
Дерево отрезков — это структура данных, которая позволяет эффективно реализовывать операции нахождения суммы/минимума/максимума элементов массива в заданном отрезке (, где l и r поступают на вход алгоритма), при этом дополнительно возможно изменение элементов массива: как изменение значения одного элемента, так и изменение элементов на целом подотрезке массива

**Структура дерева отрезков**

Подсчитаем и запомним где-нибудь сумму элементов всего массива, т.е. отрезка. Также посчитаем сумму на двух половинках этого массива: и. Каждую из этих двух половинок в свою очередь разобьём пополам, посчитаем и сохраним сумму на них, потом снова разобьём пополам, и так далее, пока текущий отрезок не достигнет длины 1. Иными словами, мы стартуем с отрезка и каждый раз делим текущий отрезок надвое (если он ещё не стал отрезком единичной длины), вызывая затем эту же процедуру от обеих половинок; для каждого такого отрезка мы храним сумму чисел на нём.

Можно говорить, что эти отрезки, на которых мы считали сумму, образуют дерево: корень этого дерева — отрезок , а каждая вершина имеет ровно двух сыновей. Отсюда и происходит название — "дерево отрезков".

Высота дерева отрезков есть величина O (log n) — например, потому что длина отрезка в корне дерева равна n, а при переходе на один уровень вниз длина отрезков уменьшается примерно вдвое.



**Реализация**

Основной реализационный момент — это то, как хранить дерево отрезков в памяти. В целях простоты мы не будем хранить дерево в явном виде, а воспользуемся таким трюком: скажем, что корень дерева имеет номер 1, его сыновья — номера 2 и 3, их сыновья — номера с 4 по 7, и так далее. Легко понять корректность следующей формулы: если вершина имеет номер i, то пусть её левый сын — это вершина с номером 2i, а правый — с номером 2i+1.

Такой приём значительно упрощает программирование дерева отрезков, — теперь нам не нужно хранить в памяти структуру дерева отрезков, а только лишь завести какой-либо массив для сумм на каждом отрезке дерева отрезков.

Стоит только отметить, что размер этого массива при такой нумерации надо ставить не 2n, а 4n. Дело в том, что такая нумерация не идеально работает в случае, когда n не является степенью двойки — тогда появляются пропущенные номера, которым не соответствуют никакие вершины дерева (фактически, нумерация ведёт себя подобно тому, как если бы n округлили бы вверх до ближайшей степени двойки). Это не создаёт никаких сложностей при реализации, однако приводит к тому, что размер массива надо увеличивать до 4n.

Итак, дерево отрезков мы храним просто в виде массива t[], размера вчетверо больше размера n входных данных: