# Алгоритмы и Структуры Данных. Лекция 1

07.02.2023

\_scarleteagle

imkochelorov

На следующей лекции будем пропихивать в детей

— Первеев Михаил Валерьевич

# Дерево отрезков (Segment tree)

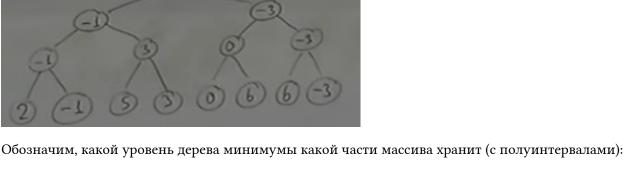
## Вступление

Пусть имеется массив a = [1 -2 5 8 7 3 6]Имеются запросы вида

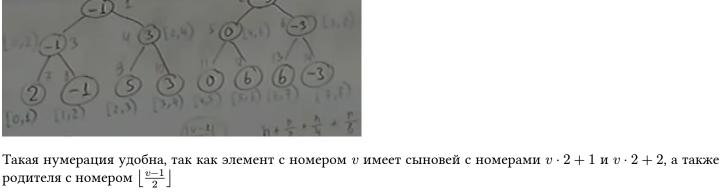
- $l,r \rightarrow \texttt{return} \ \texttt{a[1]} \ + \ \texttt{a[1+1]} \ + \ \dots \ + \ \texttt{a[r]} \ \# \ \texttt{преф}.$  суммы за O(1) часто называют RSQ (Range Summ Query) • p, x o a[p] = x # можем изменять наши преф. суммы за O(p), но потеряем всю скорость
- l, r  $\rightarrow$  return min(a[e], a[e+1], ..., a[r]) часто называют RMQ (Range Minimum Query)
- Пример:

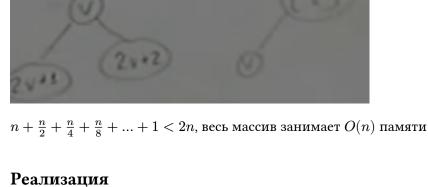
Неоходимые операции 1. min(1, r) - RMQ2. change(p, x) - RSQМассив имет вид: a = [2, -1, 5, 3, 0, 6, 6, -3]

Построим дерево, хранящее минимумы на отрезках:









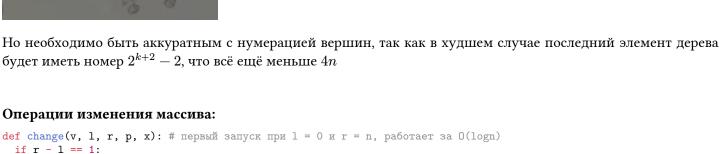
Рекурсивное построение дерева из массива:

## t # дерево отрезков

#### $def \ build(v, 1, r): \#$ первый запуск от корня до конца массива, строит дерево отрезков за O(n) $if\ r$ - l == 1: # находимся в листе

m = (1 + r) // 2# запускаемся по детям build(v \* 2 + 1, 1, m) build(v \* 2 + 1, m, r)

t[v] = min(t[v \* 2 + 1], t[v \* 2 + 2])Построение дерева отрезков корректно и в случае, когда размер массива не равен степени двойки:



t[v] = xreturn

t[v] = min(t[2 \* v + 1], t[2 \* v + 2])

Что такое плюс бесконечность в коде сами разбирайтесь, у меня доска, мне пофиг

return min(get(v \* 2 + 1, 1, m, ql, qr), get(v \* 2 + 1, m, r, ql, qr))

2) Переход:  $n \to n+1$ : т.к. у отрезка 2 конца, "худших случаев" может быть не более 2, поэтому  $\leq 4$  рек. вызовов

— Первеев Михаил Валерьевич

Для оценки времени работы докажем несколько фактов: 1.  $\forall [ql,\ qr)$  можно разбить на  $\leq 2 \cdot \log_2 n$  вершин в дереве отрезков

2. Функция get делает  $\leq 4 \cdot \log_2 n$  рекурсивных вызовов

Доказательство:

1) База: корень — 1, вызов 1 < 4

Созданная нами структура легко изменяема для выполнения запросов с любыми другими функциями, имеющими нейтральный элемент, свойства ассоциативности и "аддитивности" Задача: Дан массив. Необходимо уметь менять элемент и находить самый левый элемент, не превосходящий xРешение:

1

Построим ДО на минимум. Смотрим минимум слева от половины. Если он больше x, то ответ в левой половине,

### t[v] = a[1]return

change(v \* 2 + 2, m, r, p, x)

Пример запроса: change(2, -5)

Операция нахождения минимума на отрезке (RMQ):  $MAX_INT = 1e12$ get(v, l, r, ql, qr): if ql >= r or qr <= 1: return MAX\_INT if  $ql \le l$  and  $r \le qr$ :

return t[x] m = (1 + r) // 2

Доказательство: очевидно

Напоследок

иначе пойдем направо.