07.02.2024

_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 return a[l] + a[l + 1] + ... + a[r] # преф. суммы за 0(1) — часто называют RSQ (Range

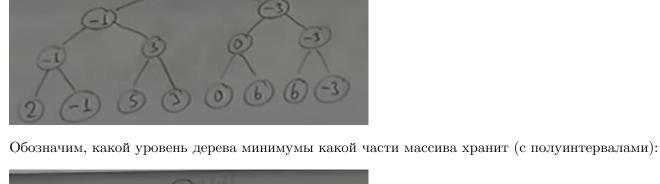
- Summ Query) • $p, x \to a[p] = x$ # можем изменять наши преф. суммы за O(p), но потеряем всю скорость
- l, r \rightarrow return min(a[e], a[e+1], ..., a[r]) часто называют RMQ (Range Minimum Query)

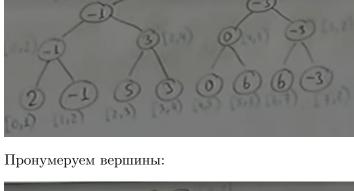
Пример:

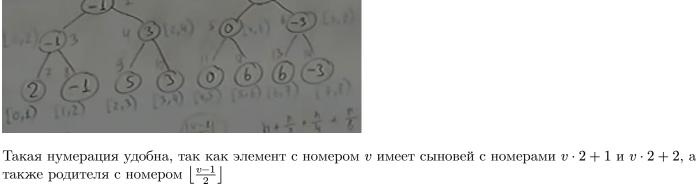
Неоходимые операции 1. min(l, r) - RMQ2. change(p, x) - RSQ

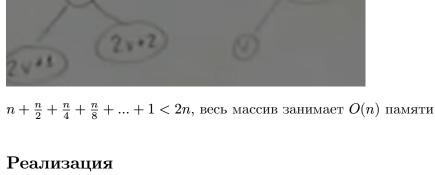
Массив имет вид: a = [2, -1, 5, 3, 0, 6, 6, -3]

Построим дерево, хранящее минимумы на отрезках:







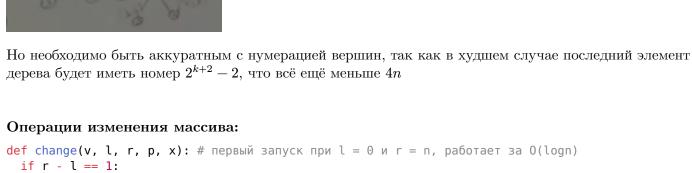


Рекурсивное построение дерева из массива: а # исходный массив

t # дерево отрезков

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

```
t[v] = a[l]
    return
  m = (l + r) // 2
  # запускаемся по детям
  build(v * 2 + 1, l, 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 m = (l + r) // 2

```
change(v * 2 + 2, m, r, p, x)
 t[v] = min(t[2 * v + 1], t[2 * v + 2])
Пример запроса:
```

change(v * 2 + 1, l, m, p, x)

if p < m:

change(2, -5)

MAX INT = 1e12

def get(v, l, r, ql, qr): if ql >= r or qr <= l:</pre> return MAX_INT if ql <= l and r <= qr:</pre>

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

Доказательство:

1) База: корень — 1, вызов 1 < 4

вызовов ⇒ ≤ 4 рек. вызовов

очевидно

```
Операция нахождения минимума на отрезке (RMQ):
```

Для оценки времени работы докажем несколько фактов:

1. $\forall [ql,\ qr)$ можно разбить на $\leq 2 \cdot \log_2 n$ вершин в дереве отрезков

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

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

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

— Первеев Михаил Валерьевич

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

Напоследок

Задача:

имеющими нейтральный элемент, свойства ассоциативности и " $a\partial\partial umu$ вности"

Дан массив. Необходимо уметь менять элемент и находить самый левый элемент, не превосходящий xПостроим ДО на минимум. Смотрим минимум слева от половины. Если он больше x, то ответ в левой половине, иначе пойдем направо.

Созданная нами структура легко изменяема для выполнения запросов с любыми другими функциями,

1