

МФТИ. ЗОШ 2025

Винер Даниил

7 января 2025 г.

План на сегодня

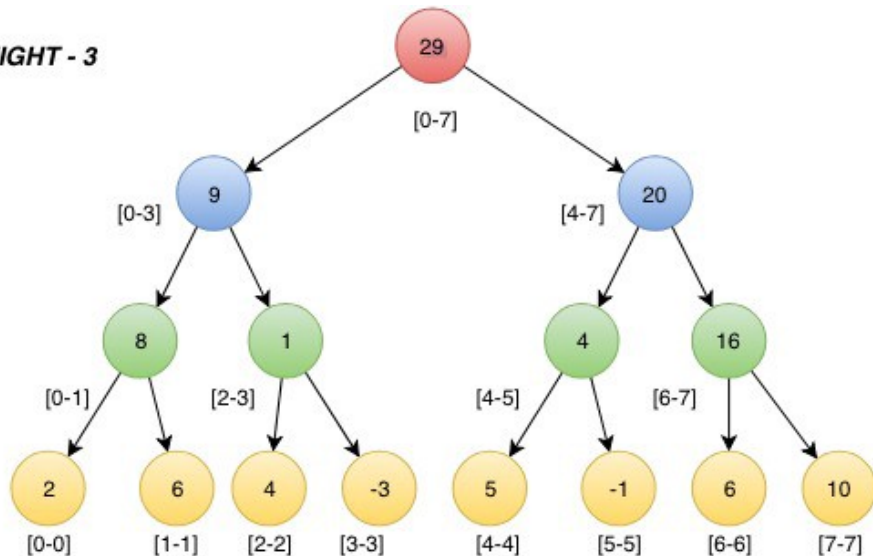
- 1 Дерево отрезков. Revision
- 2 Массовые операции
- 3 Куча
- 4 Сортировка кучей

Segment tree

- На массивах
- На указателях
- На структурах

Segment tree

HEIGHT - 3



2^0 nodes

2^1 nodes

2^2 nodes

2^3 nodes

Массовые операции на ДО

Определение

Массовые операции — операции, применяемые сразу ко всем элементам отрезка $[l; r]$

Примеры

- Сумма на отрезке, прибавление на отрезке
- Минимум на отрезке с прибавлением
- Обновление на всем отрезке

Основные идеи

- Храним два массива: *tree* и *lazy*

lazy хранит отложенные операции обновления

Прибавление на отрезке

Основные идеи

- Храним два массива: *tree* и *lazy*
lazy хранит отложенные операции обновления

Обновление отрезка за $O(\log n)$

- Если текущий узел покрывает диапазон обновления полностью, значение добавляется в массив *lazy*
- Если покрытие частичное, обновление делится между потомками

Прибавление на отрезке

Основные идеи

- Храним два массива: *tree* и *lazy*
lazy хранит отложенные операции обновления

Обновление отрезка за $O(\log n)$

- Если текущий узел покрывает диапазон обновления полностью, значение добавляется в массив *lazy*
- Если покрытие частичное, обновление делится между потомками

Ответ на запрос за $O(\log n)$

- Отложенные обновления из *lazy* применяются перед использованием узла
- Если диапазон полностью покрывает запрос, возвращается значение узла
- Иначе вычисляется максимум для левого и правого подотрезков

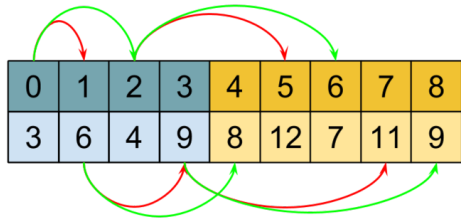
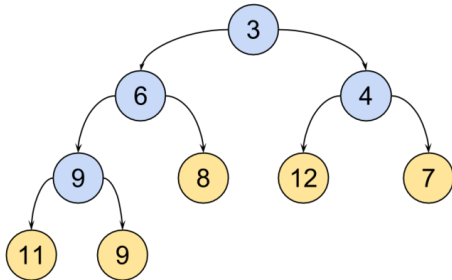
Определение

Куча — это структура данных, представляющая собой бинарное дерево, удовлетворяющее свойству кучи

- Для **max-кучи**: ключ любого узла *не меньше* ключей его потомков
- Для **min-кучи**: ключ любого узла *не больше* ключей его потомков

- $a[0]$ — элемент в корне
- На $a[2i + 1]$ и $a[2i + 2]$ позициях — потомки
- Высота — $\log n$

- $a[0]$ — элемент в корне
- На $a[2i + 1]$ и $a[2i + 2]$ позициях — потомки
- Высота — $\log n$



- Если значение измененного элемента *увеличивается*, то свойства кучи восстанавливаются функцией *siftDown*
- Если значение измененного элемента *уменьшается*, то свойства кучи восстанавливаются функцией *siftUp*

- Восстановление кучи
- Извлечение минимального/максимального элемента
- Добавление нового элемента — просеивание вверх
- Удаление элемента — провеивание вниз

- Строим max-кучу на данном массиве

- Строим max-кучу на данном массиве
- Вытаскиваем корень

- Строим max-кучу на данном массиве
- Вытаскиваем корень
- Просеиваем кучу

- Строим max-кучу на данном массиве
- Вытаскиваем корень
- Просеиваем кучу
- Повторяем процесс, пока куча не опустеет