

## Билет 20. Бинарная куча

**Основная идея:** Эффективная структура данных для очереди с приоритетом. Все операции за  $O(\log n)$ , а построение за  $O(n)$ !

## АТД Очередь с приоритетом

**Абстрактный тип данных** - интерфейс, набор операций над объектом.  
Очередь с приоритетом поддерживает:

- **Insert** - вставить элемент
- **ExtractMax** - извлечь максимум

## Бинарная куча (пирамида)

**Определение:** Подвешенное бинарное дерево со свойствами:

- **Свойство кучи:** Значение в вершине  $\geq$  значений в потомках
- **Полнота:** Все ярусы, кроме последнего, полностью заполнены
- **Упорядоченность:** Последний слой заполнен слева направо

## Хранение в массиве

Используем 0-индексацию:

- **Левый ребенок:**  $2i + 1$
- **Правый ребенок:**  $2i + 2$
- **Родитель:**  $\lfloor (i - 1) / 2 \rfloor$
- **Листья:** индексы от  $\lceil n/2 \rceil$  до  $n - 1$

## Основные операции

### SiftUp (просеивание вверх)

```
while (i > 0 and arr[i] > arr[parent(i))]:  
    swap(arr[i], arr[parent(i)])  
    i = parent(i)
```

### SiftDown (просеивание вниз)

```
while (i имеет детей и arr[i] < max(дети)):  
    j = индекс максимального ребенка  
    swap(arr[i], arr[j])  
    i = j
```

### Вставка - $O(\log n)$

1. Добавляем элемент в конец массива
2. Выполняем **SiftUp**

### Извлечение максимума - $O(\log n)$

1. Запоминаем корень (максимум)
2. Ставим последний элемент в корень
3. Выполняем **SiftDown**

### Изменение ключа - $O(\log n)$

1. Меняем значение элемента
2. Если увеличили - **SiftUp**
3. Если уменьшили - **SiftDown**

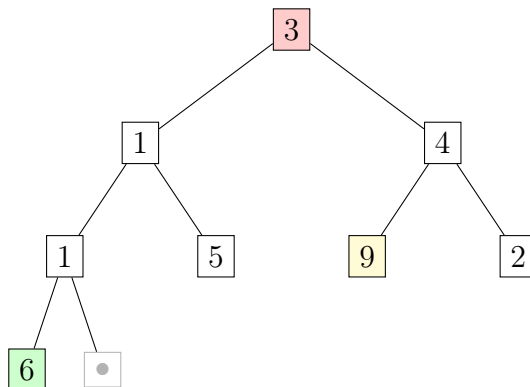
## Алгоритм построения кучи за $O(n)$

### Шаги алгоритма:

1. **Начинаем с конца:** Берём первый нелистовой элемент
2. **Идём справа налево:** От индекса  $\lfloor n/2 \rfloor - 1$  до 0
3. **Для каждого:** Выполняем SiftDown
4. **Результат:** Весь массив становится корректной кучей

### Графическая визуализация процесса

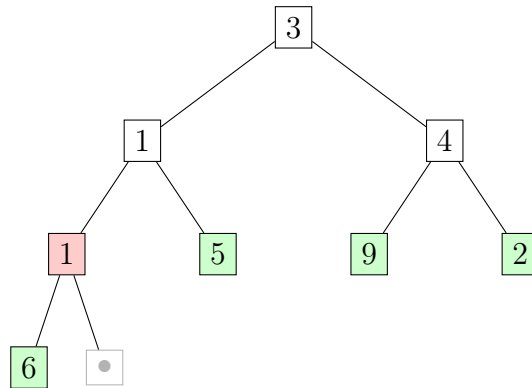
Исходный массив: [3, 1, 4, 1, 5, 9, 2, 6]



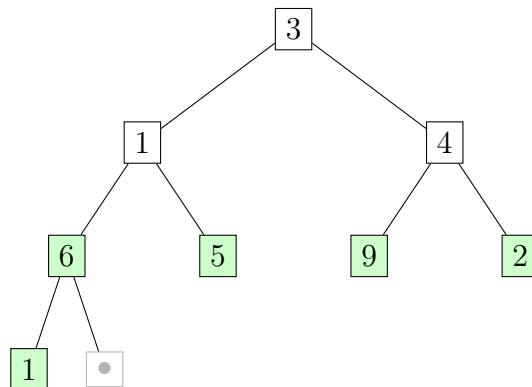
### Обозначения:

- - листья (уже корректные кучи)
- - следующий элемент для обработки
- - текущий обрабатываемый элемент

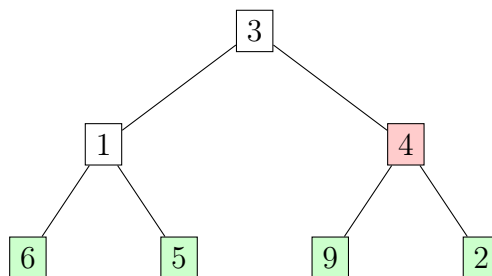
Шаг 1: Обрабатываем индекс 3 (значение 1)



SiftDown(1):  $1 < 6 \rightarrow$  меняем местами

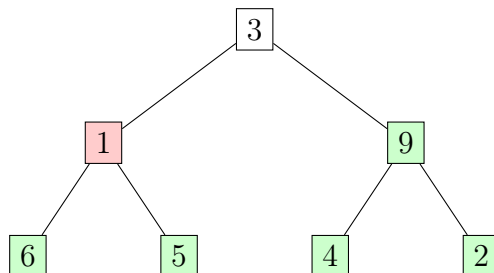


Шаг 2: Обрабатываем индекс 2 (значение 4)



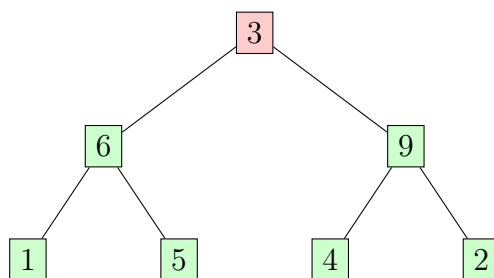
SiftDown(4):  $4 < 9 \rightarrow$  меняем местами

**Шаг 3: Обработываем индекс 1 (значение 1)**

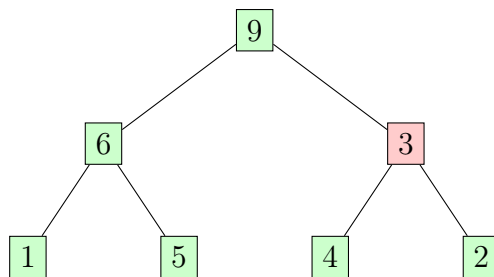


**SiftDown(1):**  $1 < \max(6,5) \rightarrow$  меняем с 6

**Шаг 4: Обработываем индекс 0 (значение 3)**

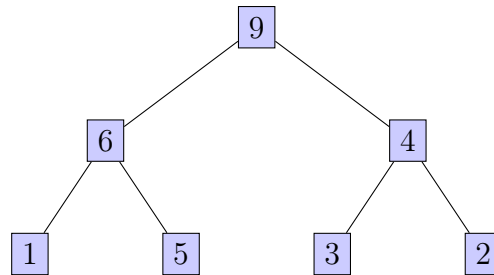


**SiftDown(3):**  $3 < \max(6,9) \rightarrow$  меняем с 9



**SiftDown(3):**  $3 < \max(4,2) \rightarrow$  меняем с 4

**Финальная куча:**



**Массив:** [9, 6, 4, 1, 5, 3, 2, 1]