

# Куча

getmin()

decreasekey()

add()

extractmin()

## Двоичная Куча

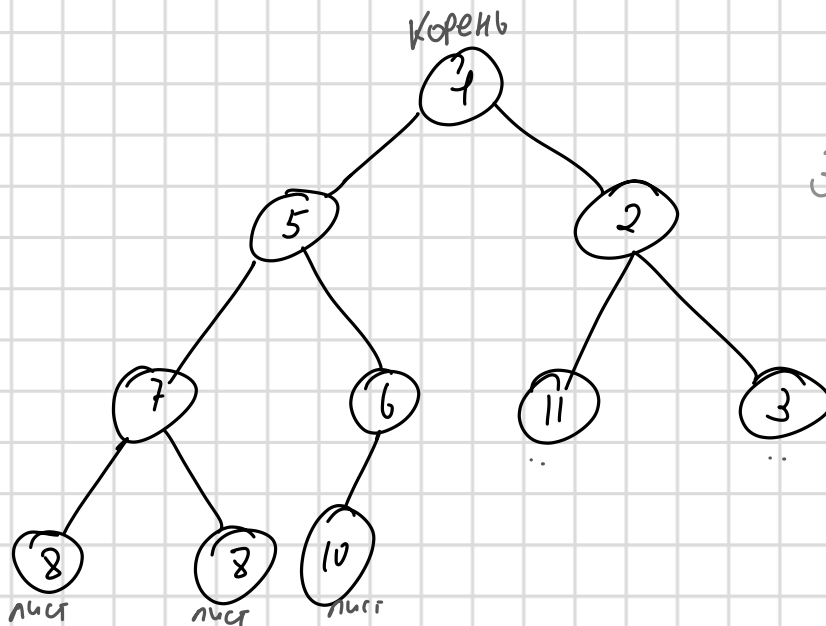
getmin()  $O(1)$

decreasekey()  $O(\log n)$

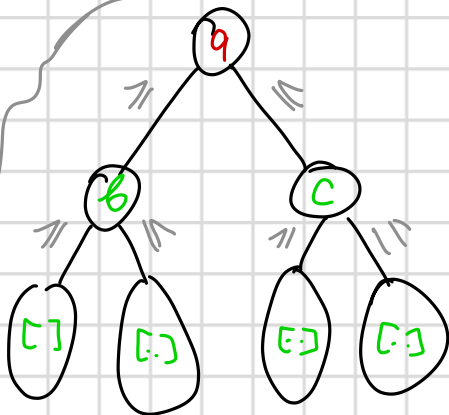
add()  $O(\log n)$

extractmin()  $O(\log n)$

получить значение вершины и нормализовать кучу



Заполнение Слева → Прав



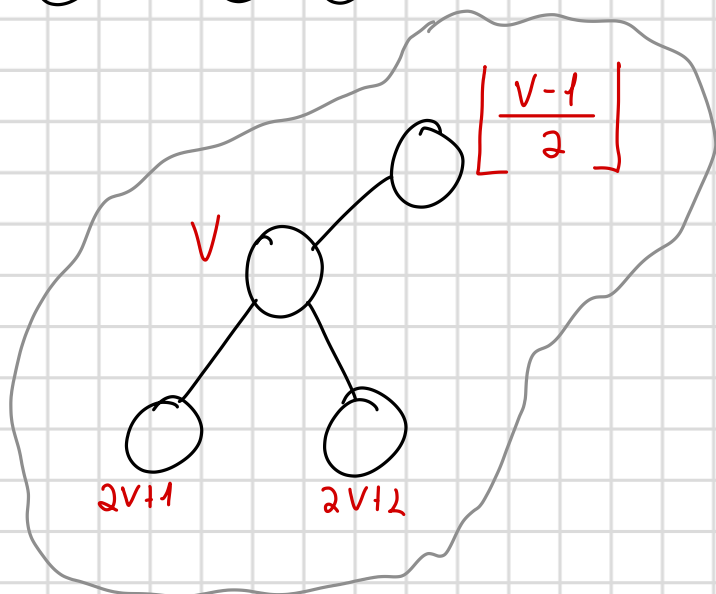
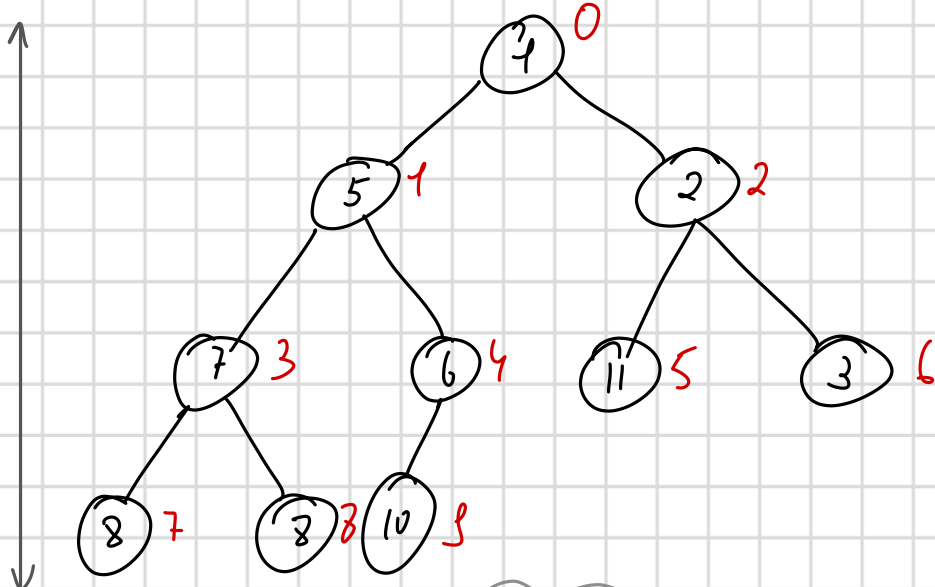
$a \leq b, c, [..]$

$b \leq [..] \dots$

Вершина всегда меньше чем все что

хранится под ней

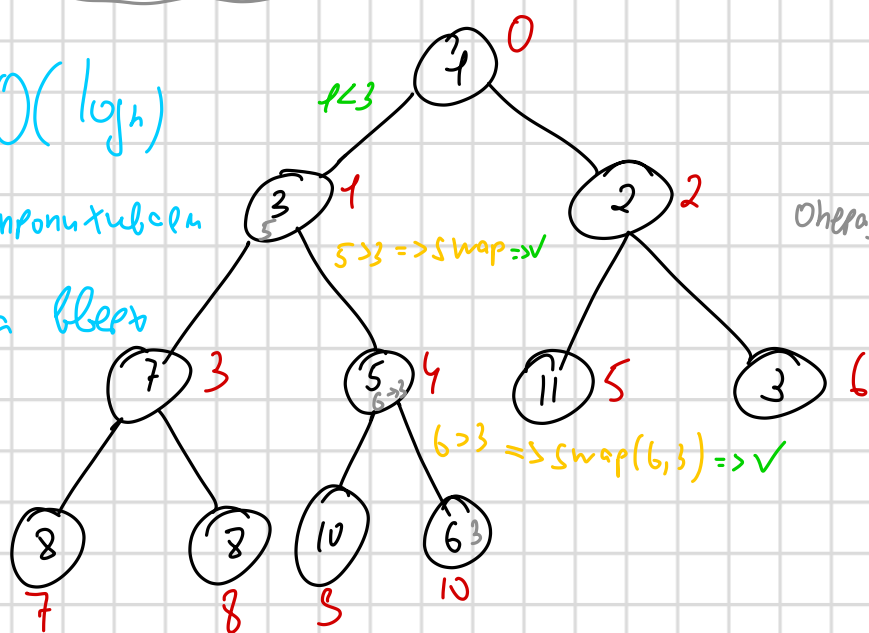
$O(\lg n)$



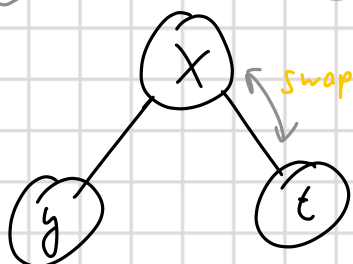
add()  $O(\lg n)$

т.к. бьеза нронтубсн

нсно с ннга бьеза

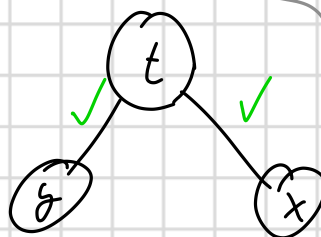


Operas. Sift up  
(нронтубсн бьеза)

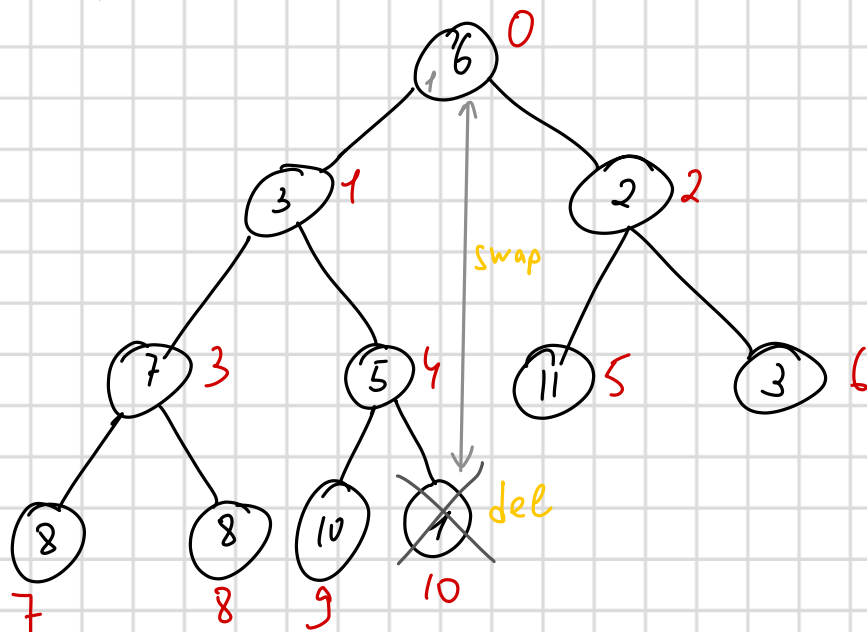


нмеем  

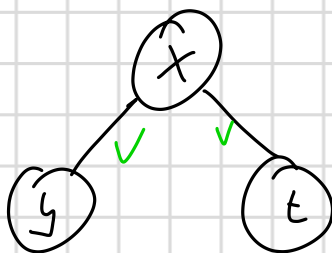
$$\left. \begin{matrix} t < x \\ x \leq y \end{matrix} \right\} t < y \longrightarrow$$



extract min()

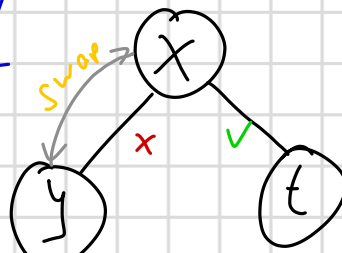


I

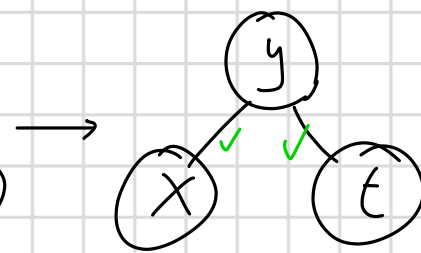


$$\begin{cases} x \leq y \\ x \leq t \end{cases}$$

II

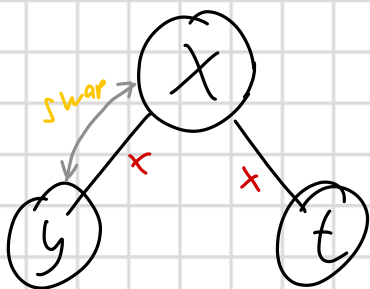


$$\begin{cases} x > y \\ x \leq t \end{cases}$$

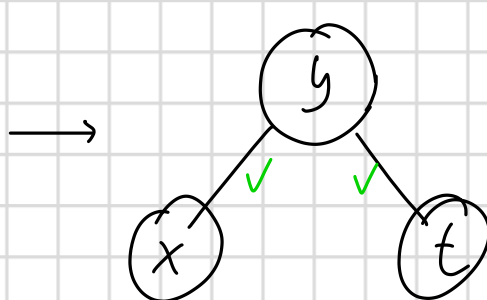


$$\begin{cases} y < x \\ x \leq t \Rightarrow y < t \end{cases}$$

III



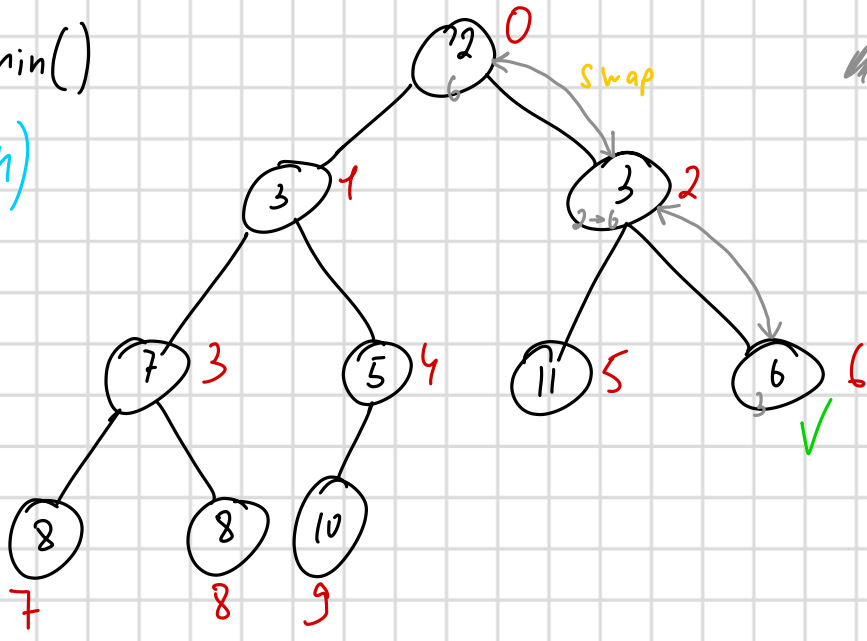
$$\begin{cases} x > y \\ x > t \\ y \leq t \Rightarrow y - \min \end{cases}$$



$$\begin{cases} x > y \\ y \leq t \end{cases}$$

но чтобы в вершину  
узел минимума среди вершин  
и детей

extract min()  
 $O(\log n)$

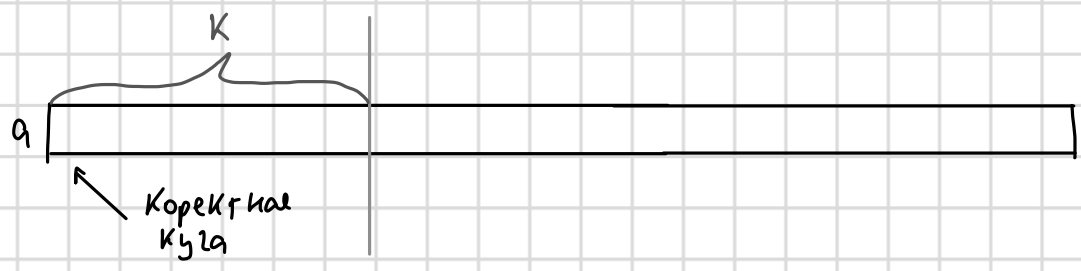
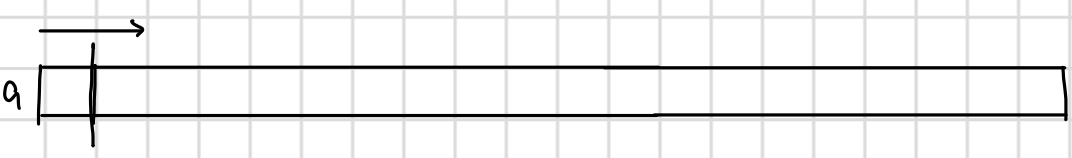


Операция  
 sift down

Реализация кучи:

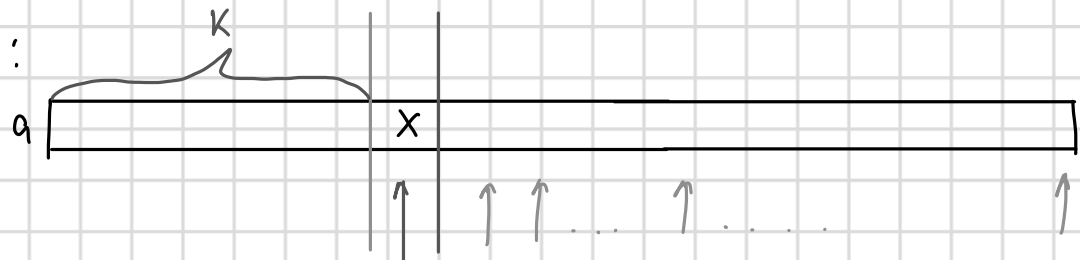
- 1) Строить кучу можно за  $O(n)$
- 2) Можно сделать  $O(1)$  для каждой

1



докажем по индукции, что при  $k=n$  весь массив станет кучей  
 База:  $a[0] \rightarrow$  какой-то корень

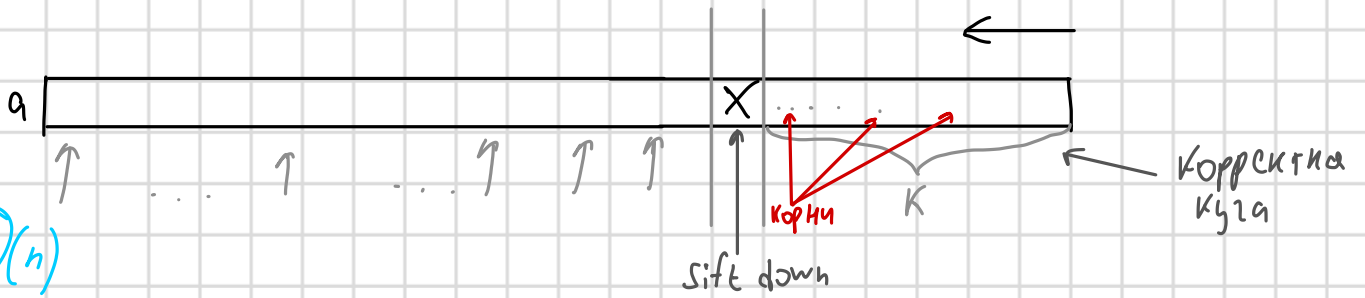
Переход:



Sift up  
 $O(\log n)$

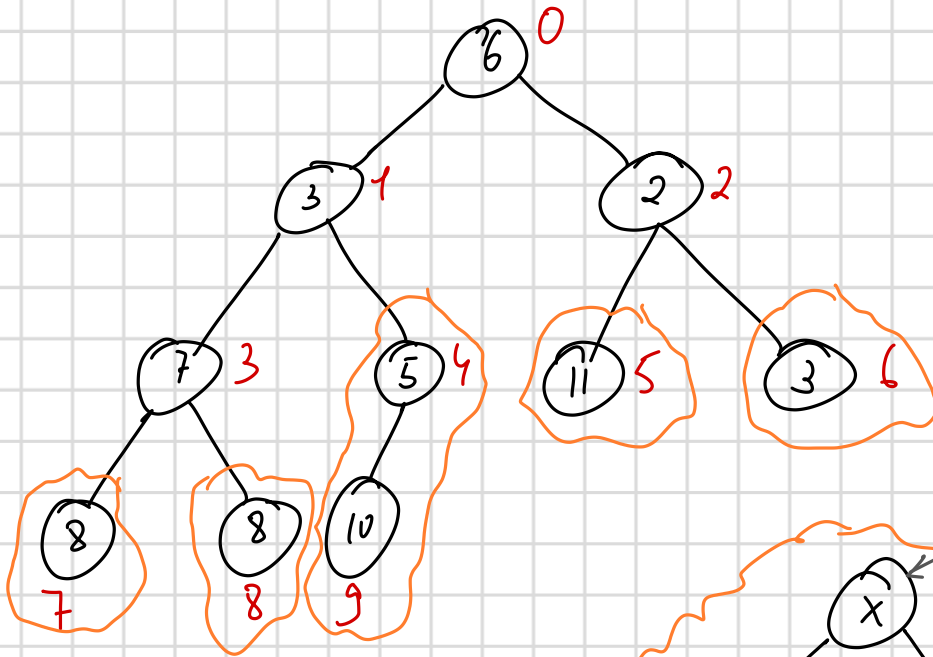
$O(n \log n)$

Хотим sift down вместо sift up

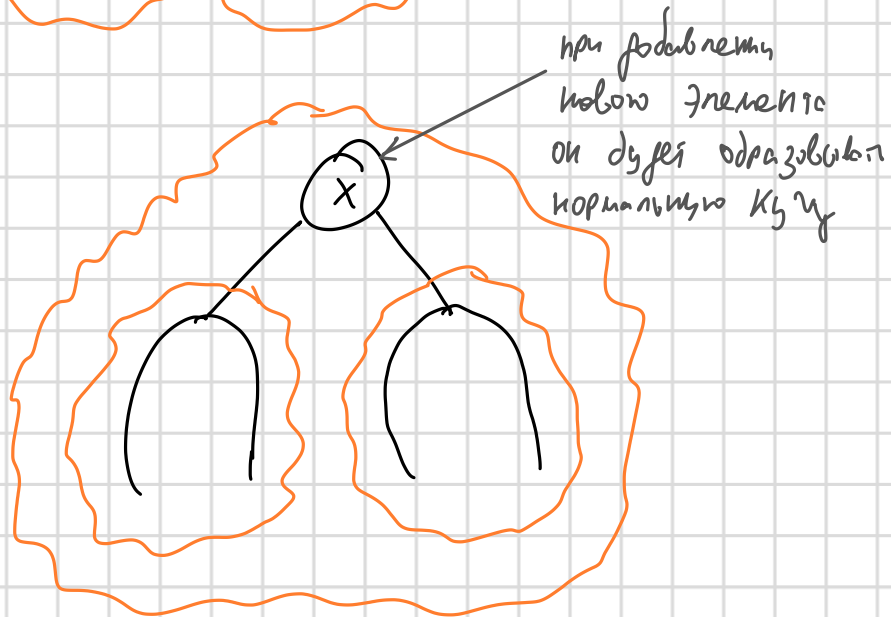


$O(n)$

Δ

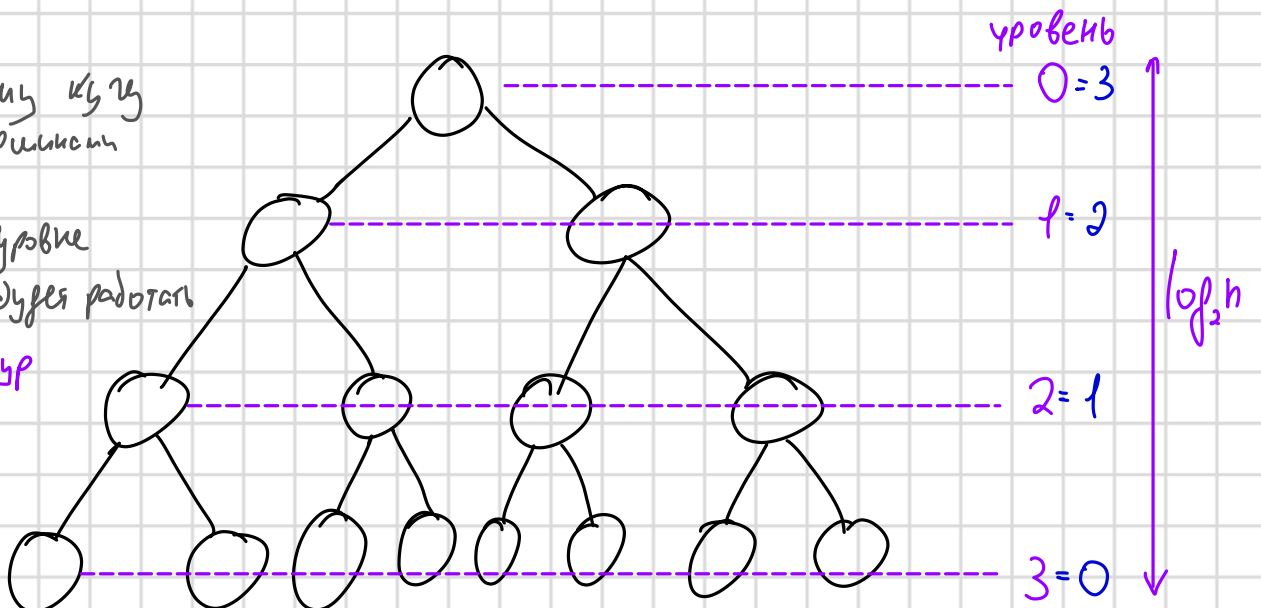


— это уже добавлено в кучу



добавим новую кучу  
пустыми вершинами

на каком уровне  
sift down должен работать  
за  $\log_2 n$  — #ур



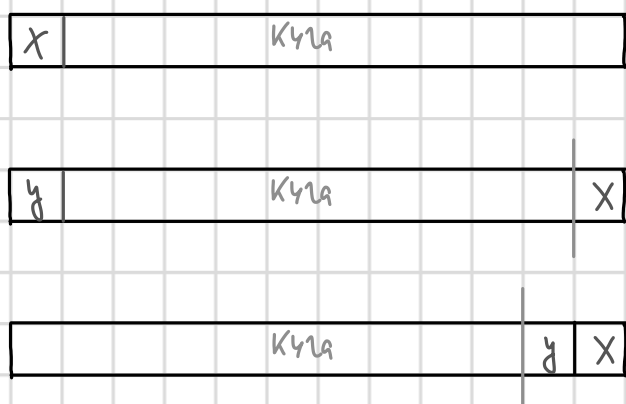
$$\sum_{i=0}^{\log_2 n} (i+1) \cdot \frac{n}{2^{i+1}} = \sum_{i=1}^{\log_2 n+1} i \cdot \frac{n}{2^i} = n \cdot \sum_{i=1}^{\log_2 n+1} \frac{i}{2^i} \leq 2 \cdot n = O(n)$$

$$\sum_{i=1}^{\infty} \frac{i}{2^i} = 2$$

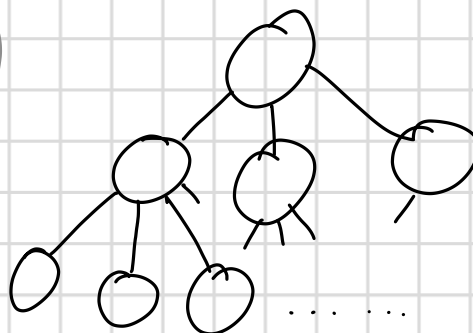
$$\begin{array}{ccc} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{16} & \vdots \\ \vdots & \vdots & \vdots \end{array} \quad \frac{1}{16} \quad \vdots$$

$\underbrace{\quad}_{=1} \quad \underbrace{\quad}_{=\frac{1}{2}} \quad \underbrace{\quad}_{=\frac{1}{4}} \quad \dots = 2$

2



K-ичная куча



Sift up  $\log_K n$  операций ↑

Sift down  $(\log_K n) \cdot K$  операций ↓

## Бинарный поиск

$$a_0 \leq a_1 \leq a_2 \leq \dots \leq a_{n-1}$$

$$\text{Min } i : a_i \geq x$$

$$l = -1 \quad r = n$$

Инвариант:

$$\begin{cases} a_l < x \\ a_r \geq x \end{cases}$$

while (r - l > 1):  $O(\log n)$

m = (r + l) / 2

if (a[m] < x):

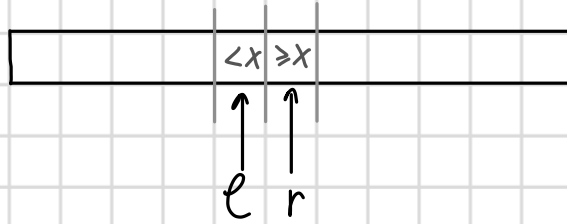
l = m

else:

r = m

lower\_bound

в итоге



## Нахождение корня монотонной ф-ции:

Хотим найти с точностью

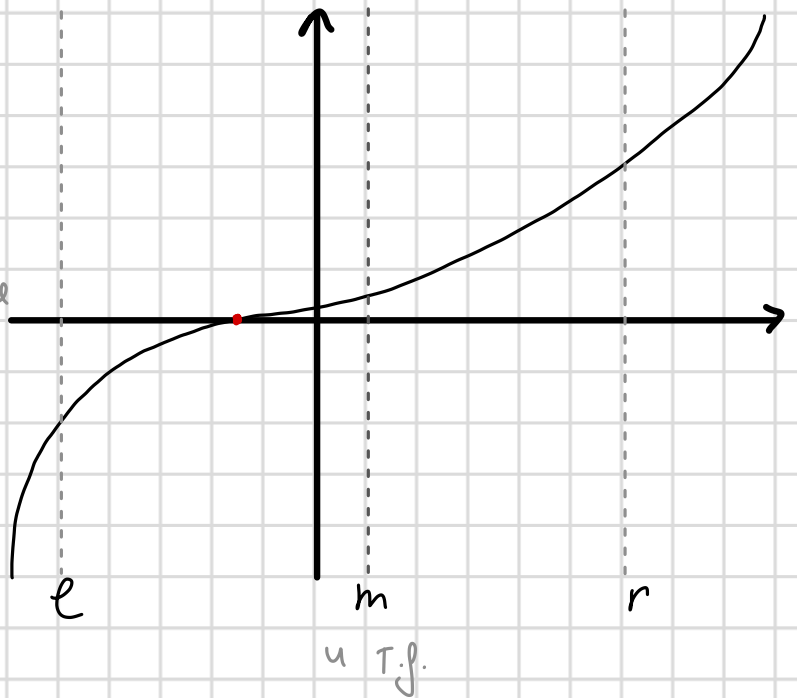
$\epsilon = 10^{-6}$

~~while (r - l >  $\epsilon$ )~~

~~m = (l + r) / 2~~

т.к. точка  
может зажаться  
при добавлении l, r

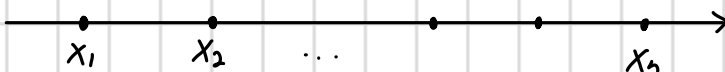
for (i = 0; i < ITERS; i++)



$O(\log \frac{r-l}{\epsilon})$

## Бинар. поиск по ответу

n элементов



$V_i$  - max скорость; min время встрет. - ?

$$f(t) = \begin{cases} 1, & \text{можно вып. за } t \\ 0, & \text{иначе} \end{cases}$$

$$f(t) = 1 \Rightarrow \forall \varepsilon > 0 \quad f(t + \varepsilon) = 1$$

$$f(t) = 0 \Rightarrow \forall \varepsilon > 0 \quad f(t - \varepsilon) = 0$$

$$l = 0$$

$$r = \frac{x_{\max} - x_{\min}}{\min V_i}$$

$$f(m)$$

$$O(n \cdot \log \frac{r-l}{\varepsilon})$$

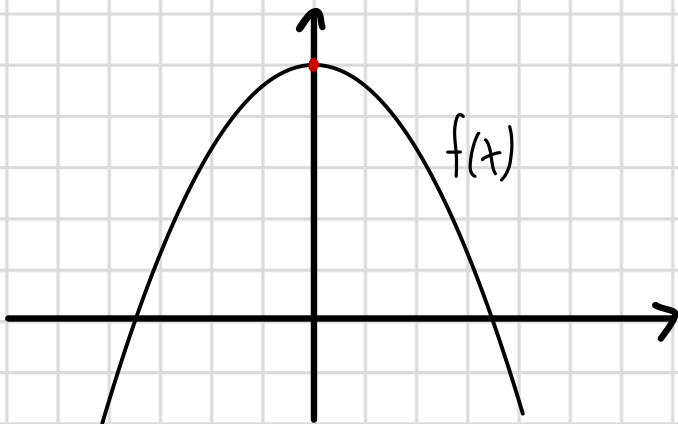
$$x_i - V_i \cdot m$$

$$x_i$$

$$x_i + V_i \cdot m$$

$$[l_1, r_1] \cap [l_2, r_2] = [\max(l_1, l_2), \min(r_1, r_2)]$$

## Тернарный поиск



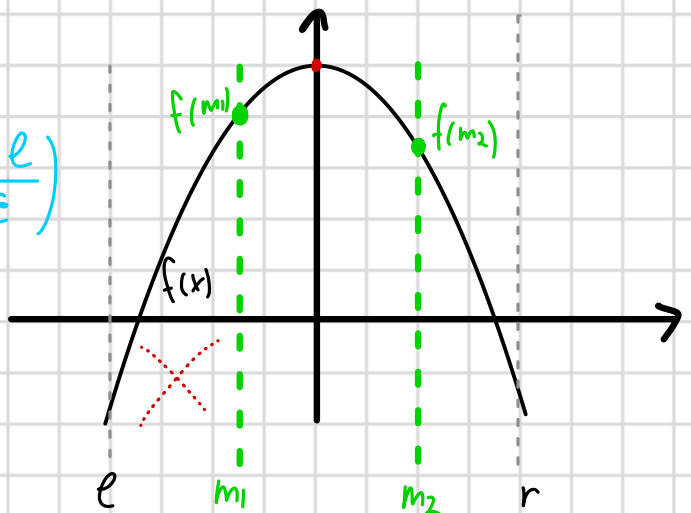
$$f'(x) = 0$$

использовать и через бинар. поиск можно

$$f(m_1) < f(m_2)$$

$$l \leftarrow m_1$$

$$O(\log_{1.5} \frac{r-l}{\varepsilon})$$



c f:

