Куча

getmin()   decreaseKey()

add()

extractmin()

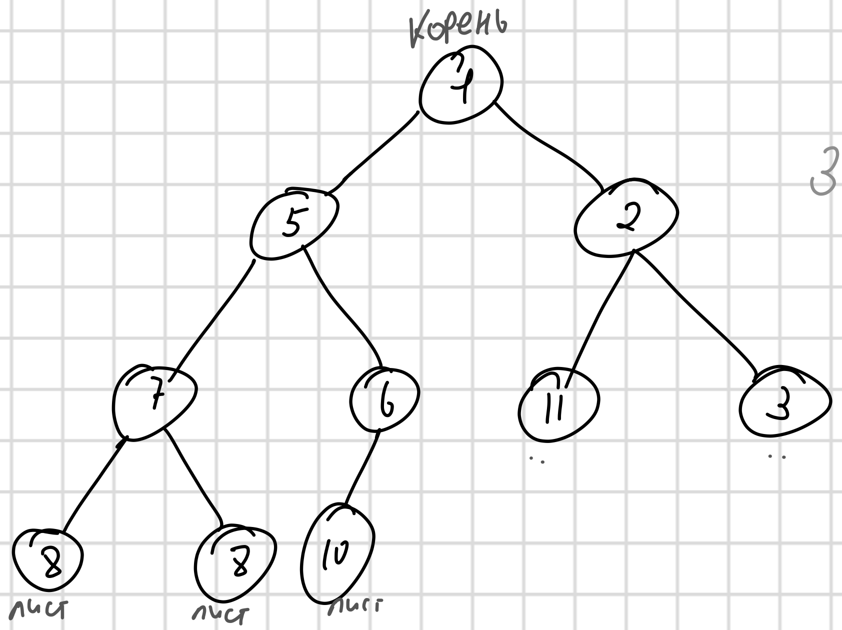
Двоичная куча

getmin() $O(1)$ decreaseKey() $O(\log n)$

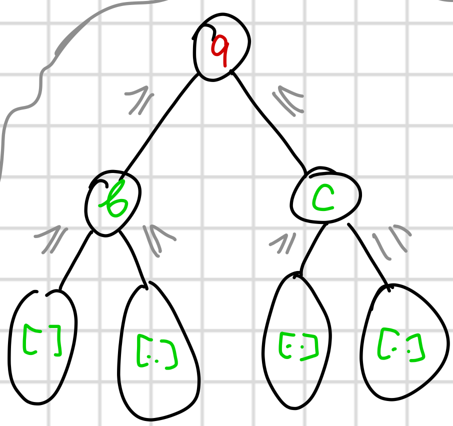
понизить значение вершины и нормализовать кучу

add() $O(\log n)$

extractmin() $O(\log n)$



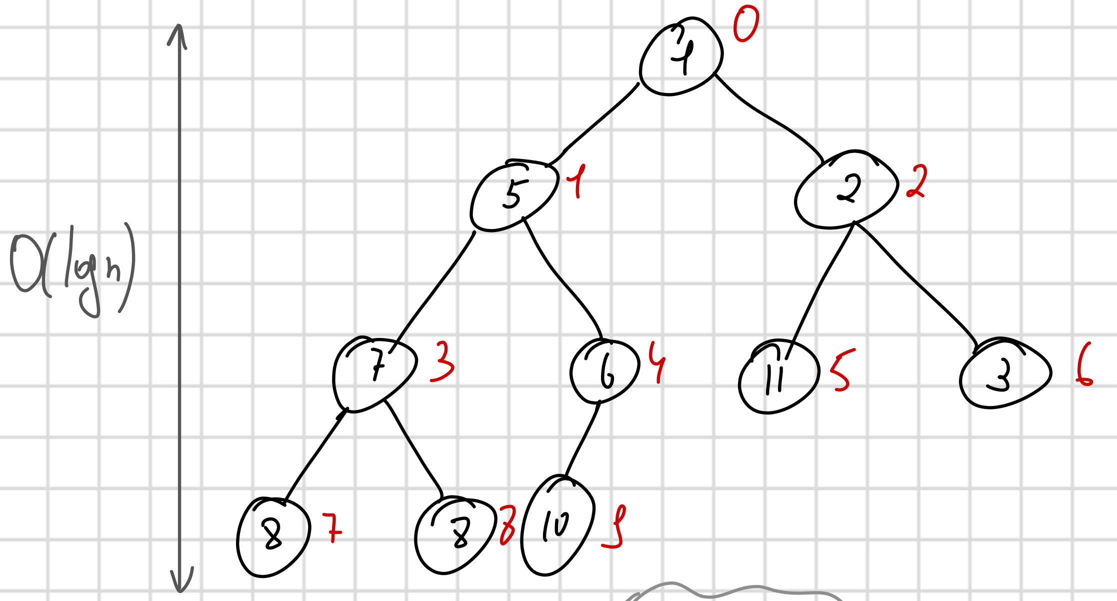
заполнение слева→направо

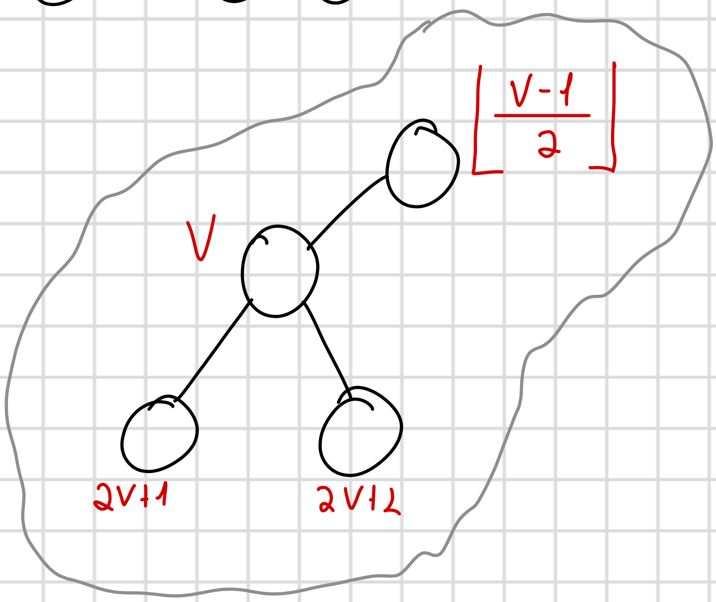


$a \le b, c, [...]$

$b \le [...]$ ...

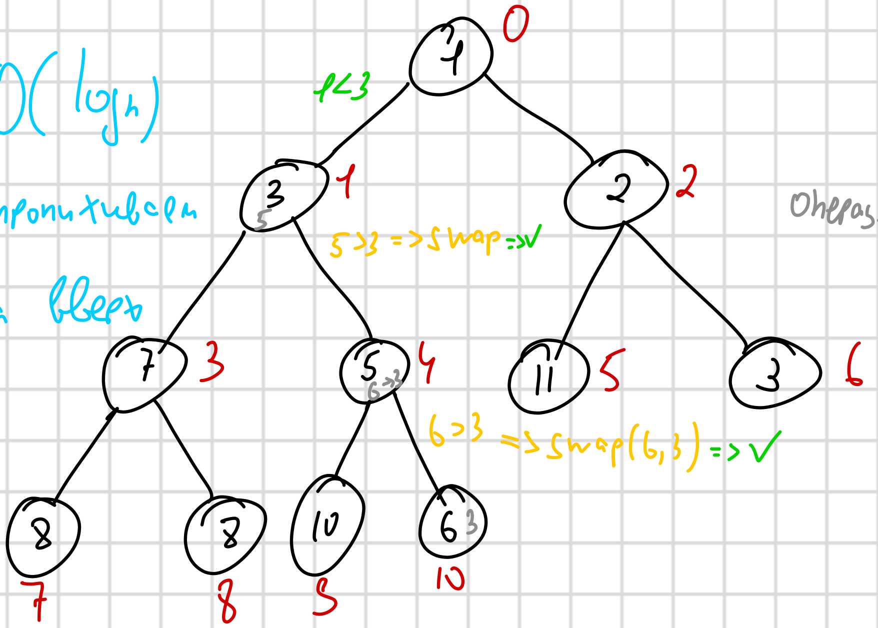
вершина всегда меньше чем все что хранится под ней



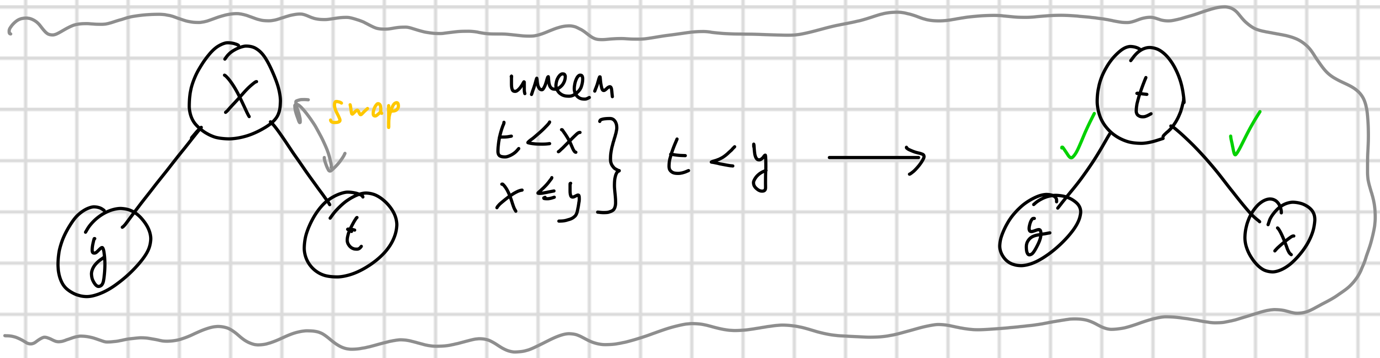


add() O(log n)

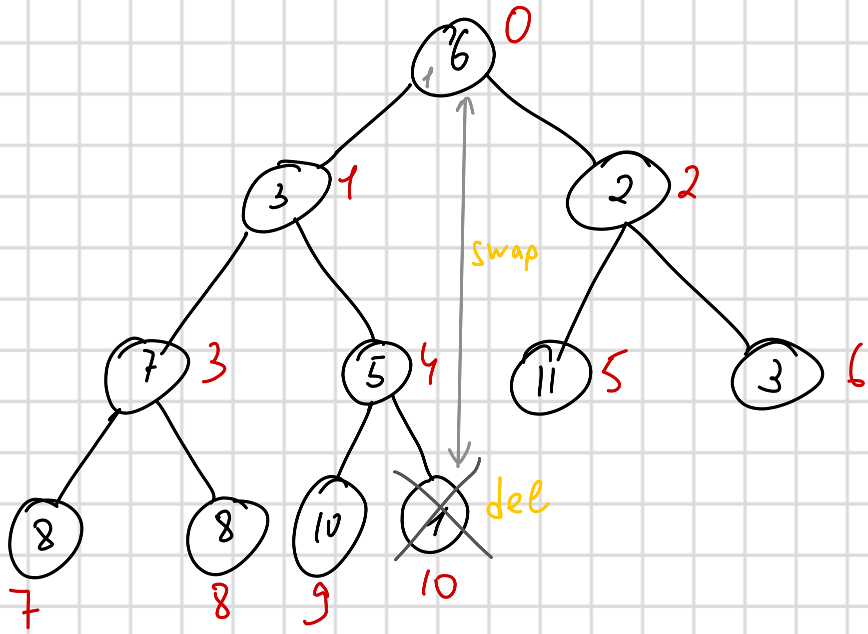
т.к. всегда протискиваем число с низа вверх



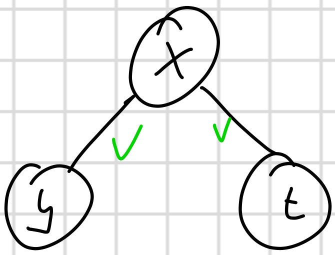
Операц. sift up (просачивание вверх)



extract min()

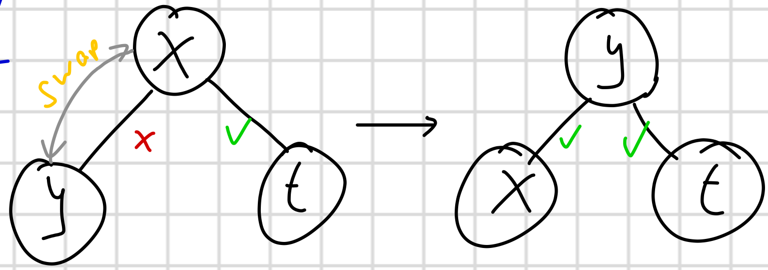


I



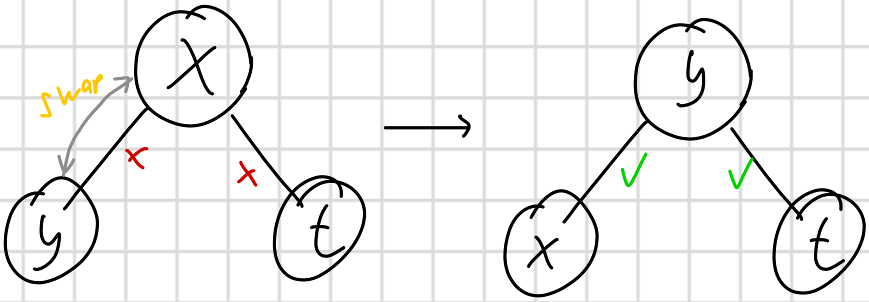
$$\begin{cases} x \le y \\ x \le t \end{cases}$$

II



$$\begin{cases} x > y \\ x \le t \end{cases} \qquad \begin{cases} y < x \\ x \le t \Rightarrow y < t \end{cases}$$

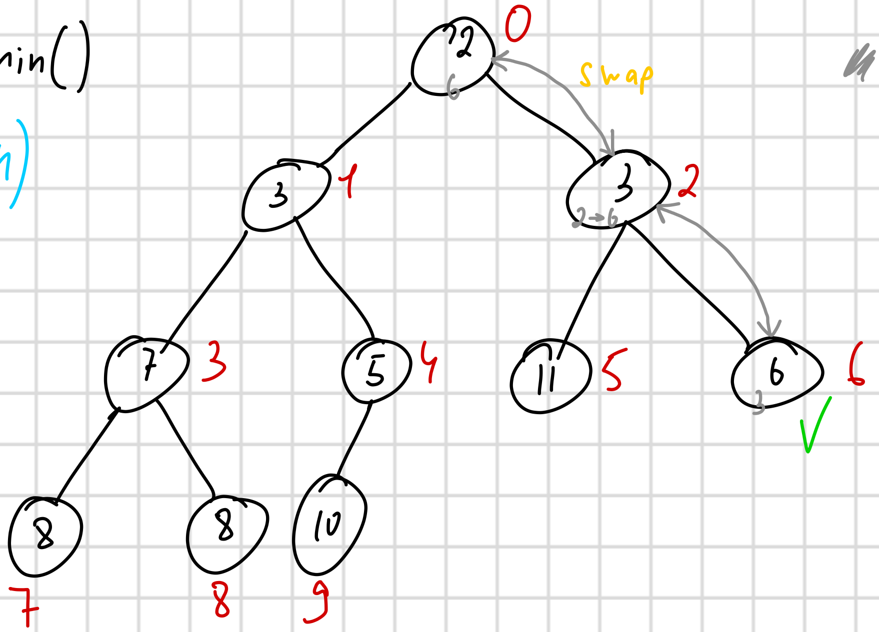
III



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

по итогу в вершину идет минимум среди вершины и детей extract min()

$O(\log n)$

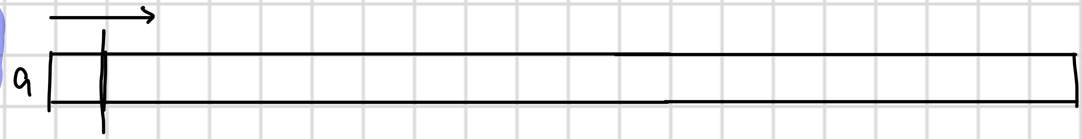


Операции sift down

Реализация кучи:

1) Строить кучу можно за $O(n)$
2) Можно сделать $O(1)$ доп памяти

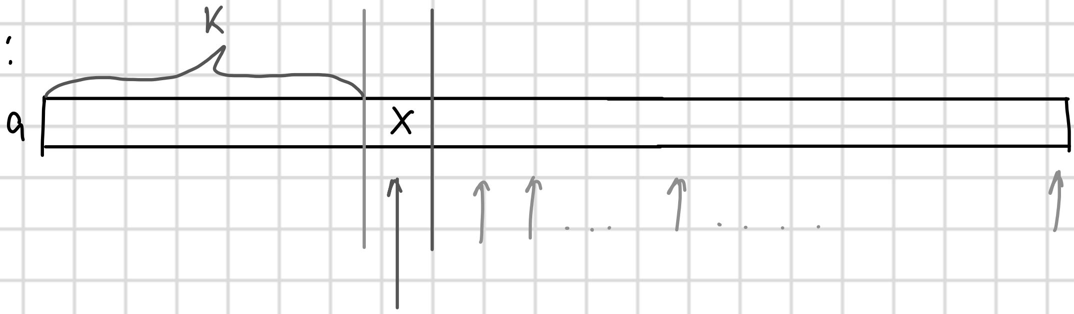
1



Докажем по индукции, что при $k=n$ весь массив станет кучей

База: $\exists\ a[0] \to$ какой-то корень

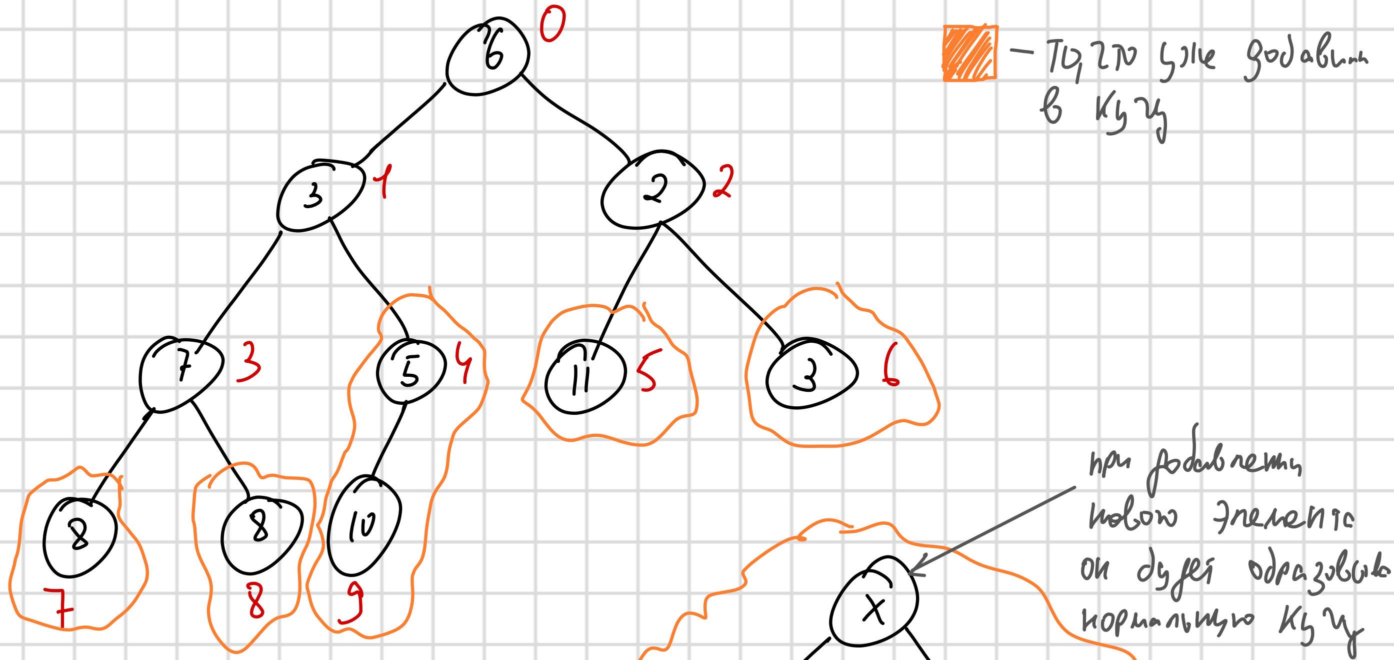
Переход:

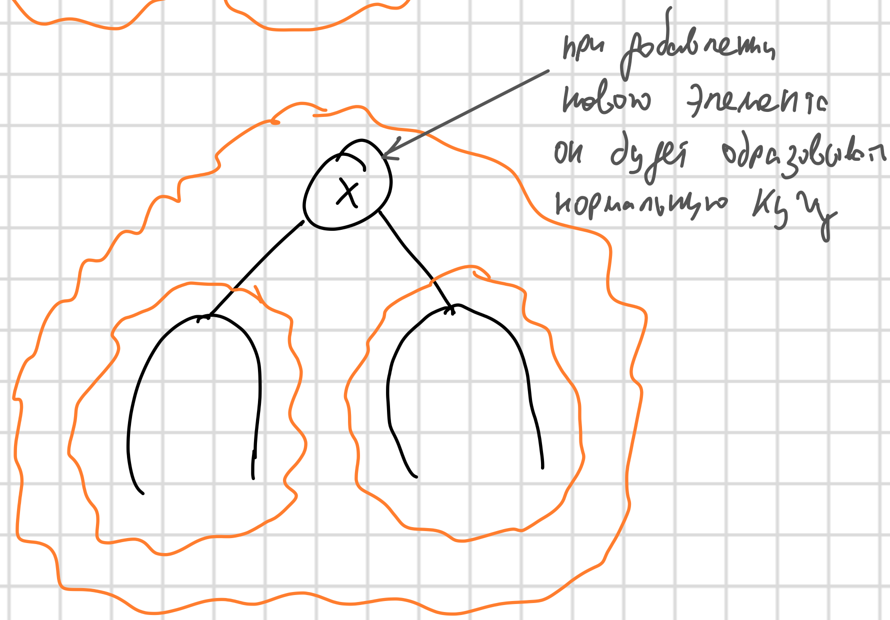


sift up $O(\log n)$

$O(n \log n)$

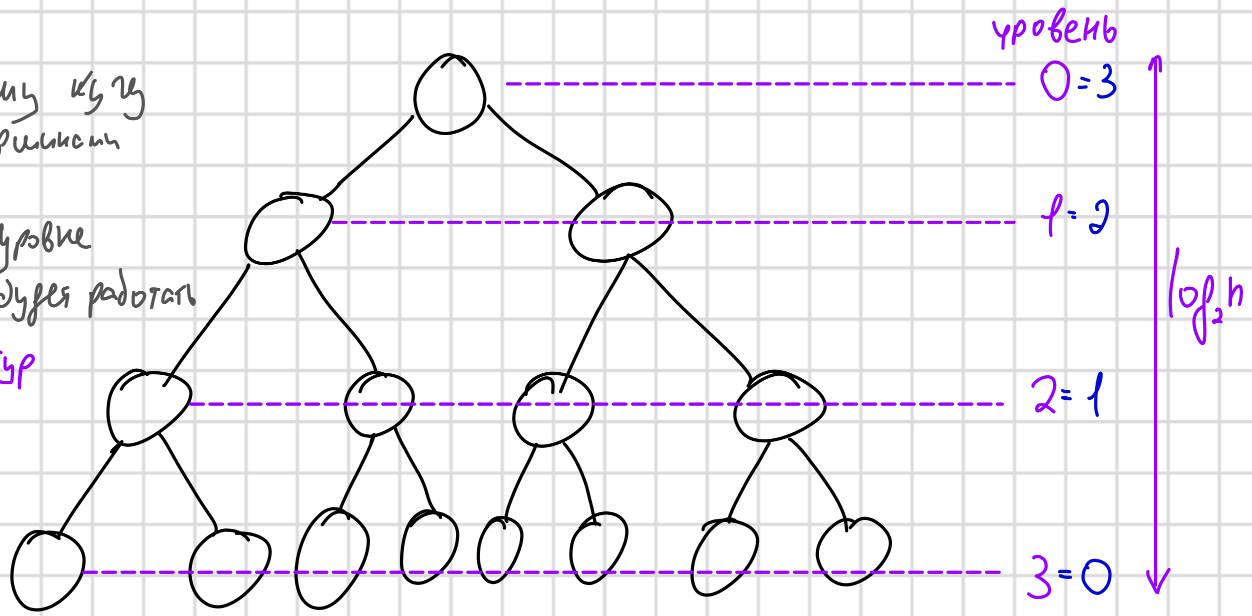
хотим sift down вместо sift up





добьем нашу кучу пустыми вершинами

на каждом уровне sift down будет работать за $\log_2 n - \#\text{ур}$



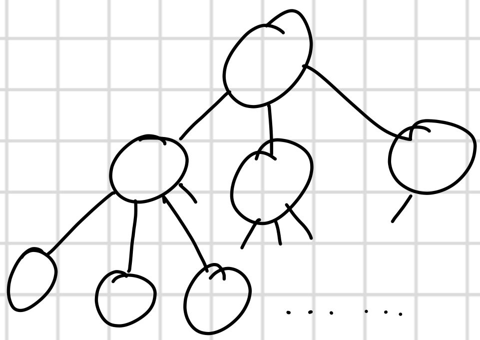
$$\sum_{i=0}^{\log_2 n} (i+1)\cdot\frac{n}{2^{i+1}} = \sum_{i=1}^{\log_2 n+1} i\cdot\frac{n}{2^i} = n\cdot\sum_{i=1}^{\log_2 n+1}\frac{i}{2^i} \le 2\cdot n = O(n)$$

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

$$\begin{array}{cccc}
\frac{1}{2} & & & \\
\frac{1}{4} & \frac{1}{4} & & \\
\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
\vdots & \vdots & \vdots & \vdots \\
\parallel & \parallel & \parallel & \\
1 & \frac{1}{2} & \frac{1}{4} & \dots = 2
\end{array}$$

2

K-ичная куча



sift up $\log_k n$ скорость ↑

sift down $(\log_k n)\cdot k$ скорость ↓

Бинарный поиск

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

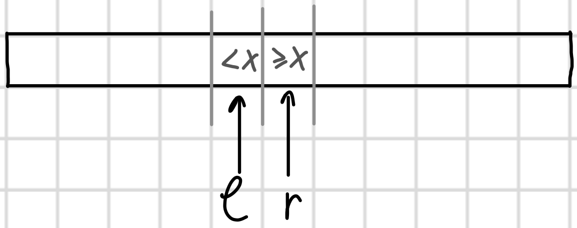
Min i : $a_i \ge x$

$l = -1 \qquad r = n$

Инвариант: $\begin{cases} a_l < x \\ a_r \ge x \end{cases}$

```
while (r-l>1):   O(log n)
    m = (r+l)/2
    if (a[m] < x):
        l = m
    else:
        r = m
```

В итоге



$\boxed{\text{lower\_bound}}$

Нахождение корня монотонной ф-ии:

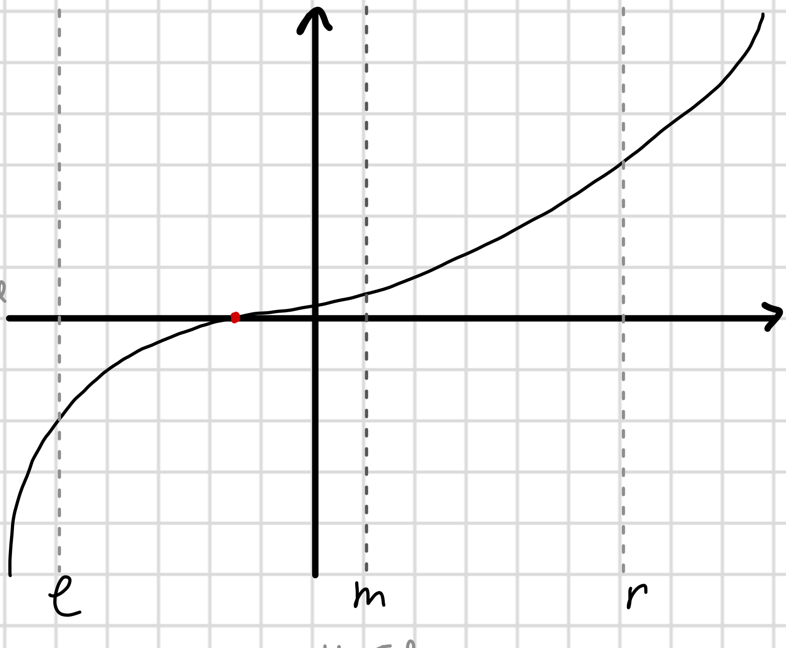
Хотим сходится с точностью

$\varepsilon = 10^{-6}$

```
~~while (r-l > ε)~~
~~    m = (l+r)/2~~
      ⋮
```

т.к. цикл может зациклиться при больших l, r

```
for (i=0; i<ITERS; i++)
    ⋮
```

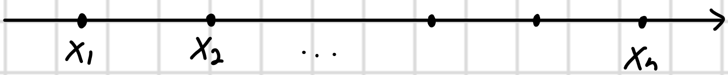


и т.д.

$O\left(\log \frac{r-l}{\varepsilon}\right)$

Бин. поиск по ответу

n человек



V_i – max скорость; min время встречи – ?

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

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

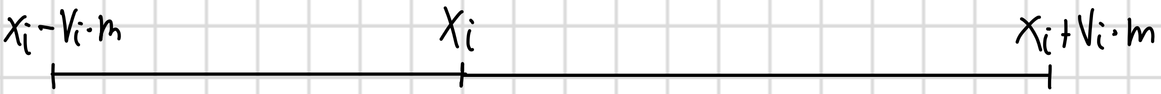
$$f(t) = 0 \Rightarrow \forall \varepsilon > 0 \quad f(t-\varepsilon) = 0$$

$$l = 0$$

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

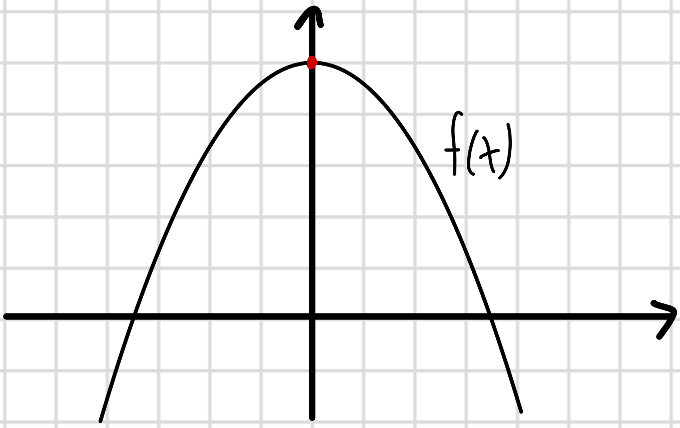
$$O\left(n \cdot \log \frac{r-l}{\varepsilon}\right)$$

$$f(m)$$



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

Тернарный поиск



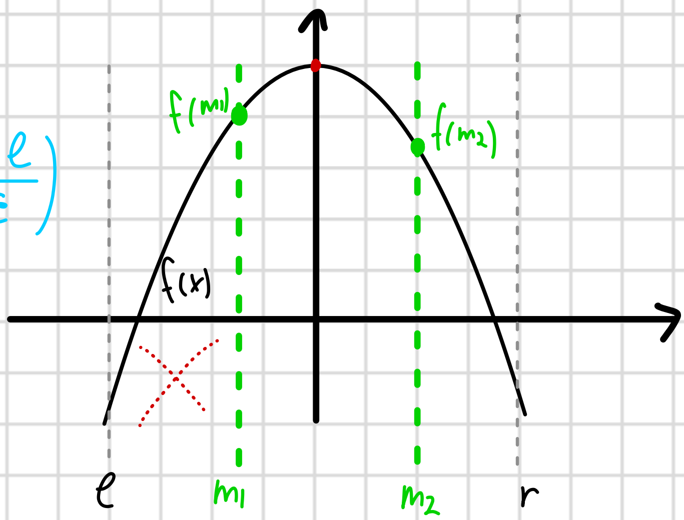
$$f'(x) = 0$$

посчитать и через бин. поиск най-ти

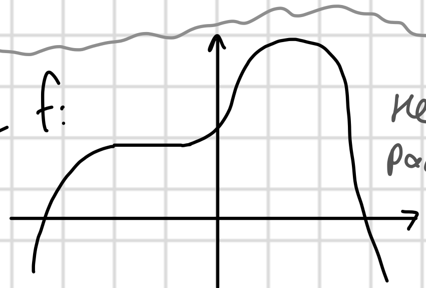
$$f(m_1) < f(m_2)$$

$$l = m_1$$

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



С f:



не работает