

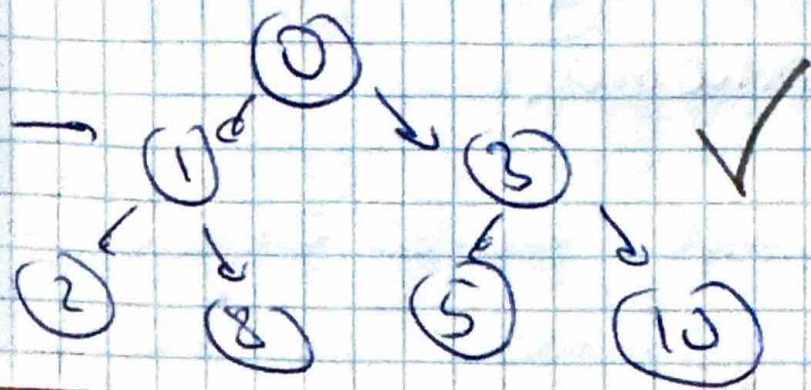
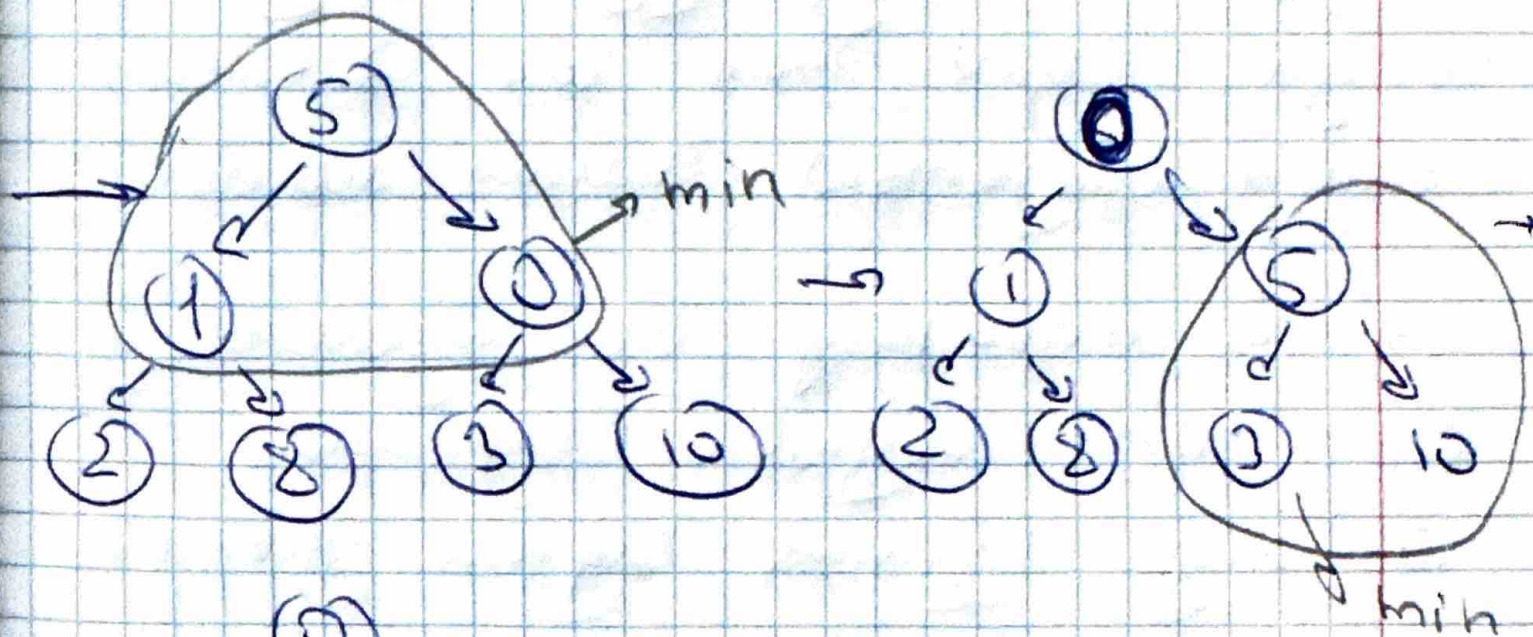
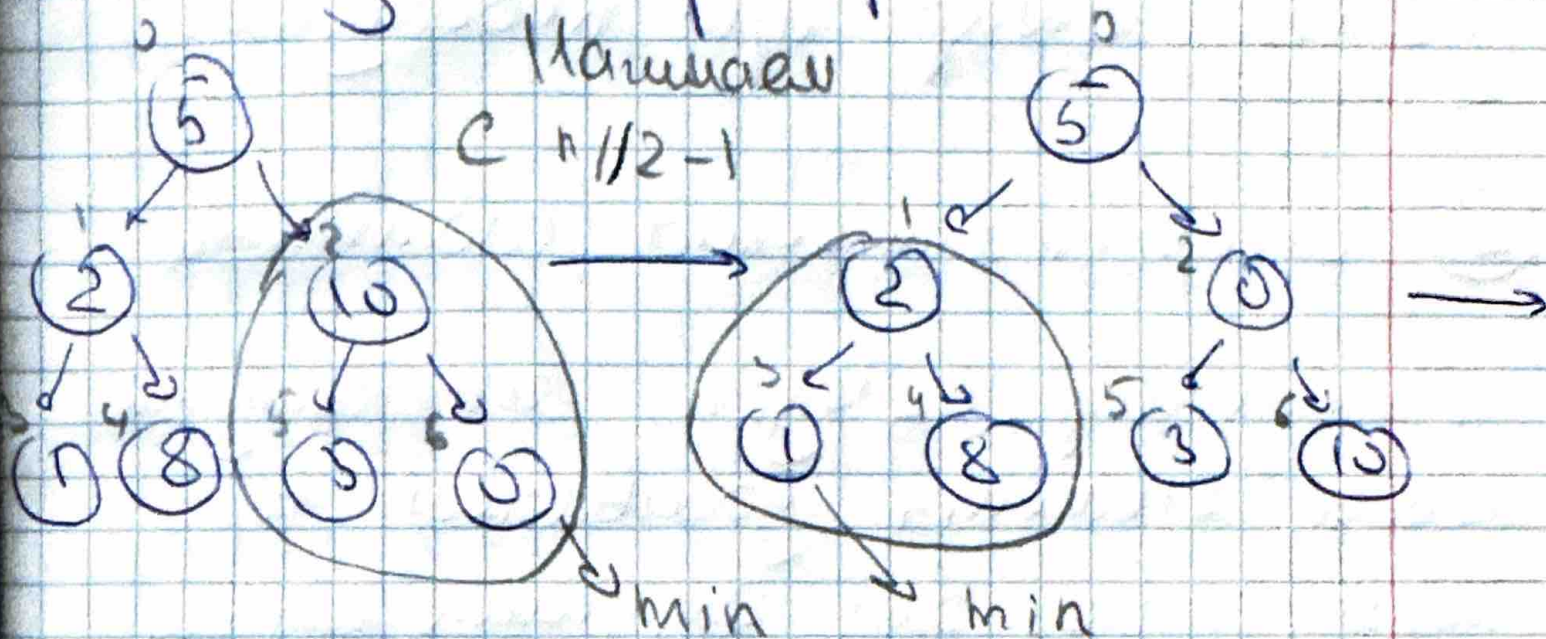
# Функция MakeHeap $O(n)$

① Подход к решению

Показываю на примере:

Начинаю

с  $n/2 - 1$



Мы перебираем элементы от самого  
правого роутера. От текущего  
роутера и вниз среди висящих  
тройки вкбараем минимальный,  
делаем swap при необходимости.

② Асимптотическая сложность.

В makeheap -  $\log n$  и мы каждый  
новый элемент добавляем в  
конечу. Поэтому худший случай  
всегда  $\log N$ , так как элемент  
может оказаться самым миним.

НО в makeheap мы уже всегда  
вниз от текущего роутера.  
Поэтому худший случай зависит  
от текущей высоты.

Пусть  $T(n)$  - общее кол-во операций,  
выполняемое алгоритмом при об-

работе и операции.

$h$  - высота дерева,  $l$  - высота текущей  
уровня, тогда

$$T(N) = \sum_{l=0}^{h-1} 2^l (h-l)$$

Заменим

$$k = h - l \quad \text{тогда} \quad l = h - k$$

$$\begin{aligned} T(N) &\leq \sum_{k=1}^h 2^{h-k} k = \sum_{k=1}^h 2^h \cdot 2^{-k} \cdot k = \\ &= \sum_{k=1}^h 2^h \cdot \frac{k}{2^k} = 2^h \sum_{k=1}^h \frac{k}{2^k} \end{aligned}$$

Зная, что  $\sum_{k=1}^{\infty} \frac{k}{2^k} = 2$ , получим

$$T(N) \leq 2^h \cdot 2 = 2N \quad 2^h \leq N \text{ по условию}$$

$$T(N) \leq 2N \Rightarrow O(n)$$