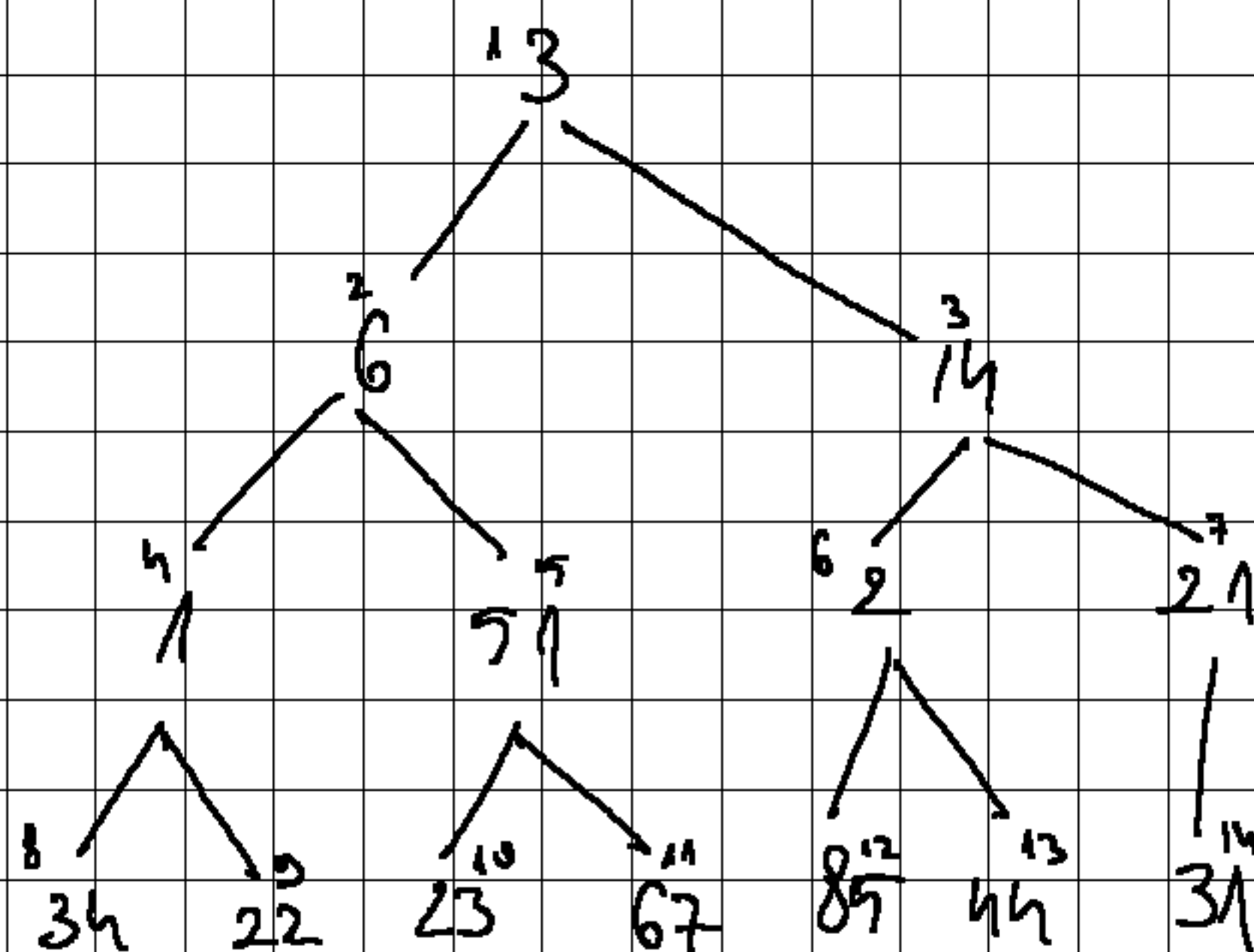
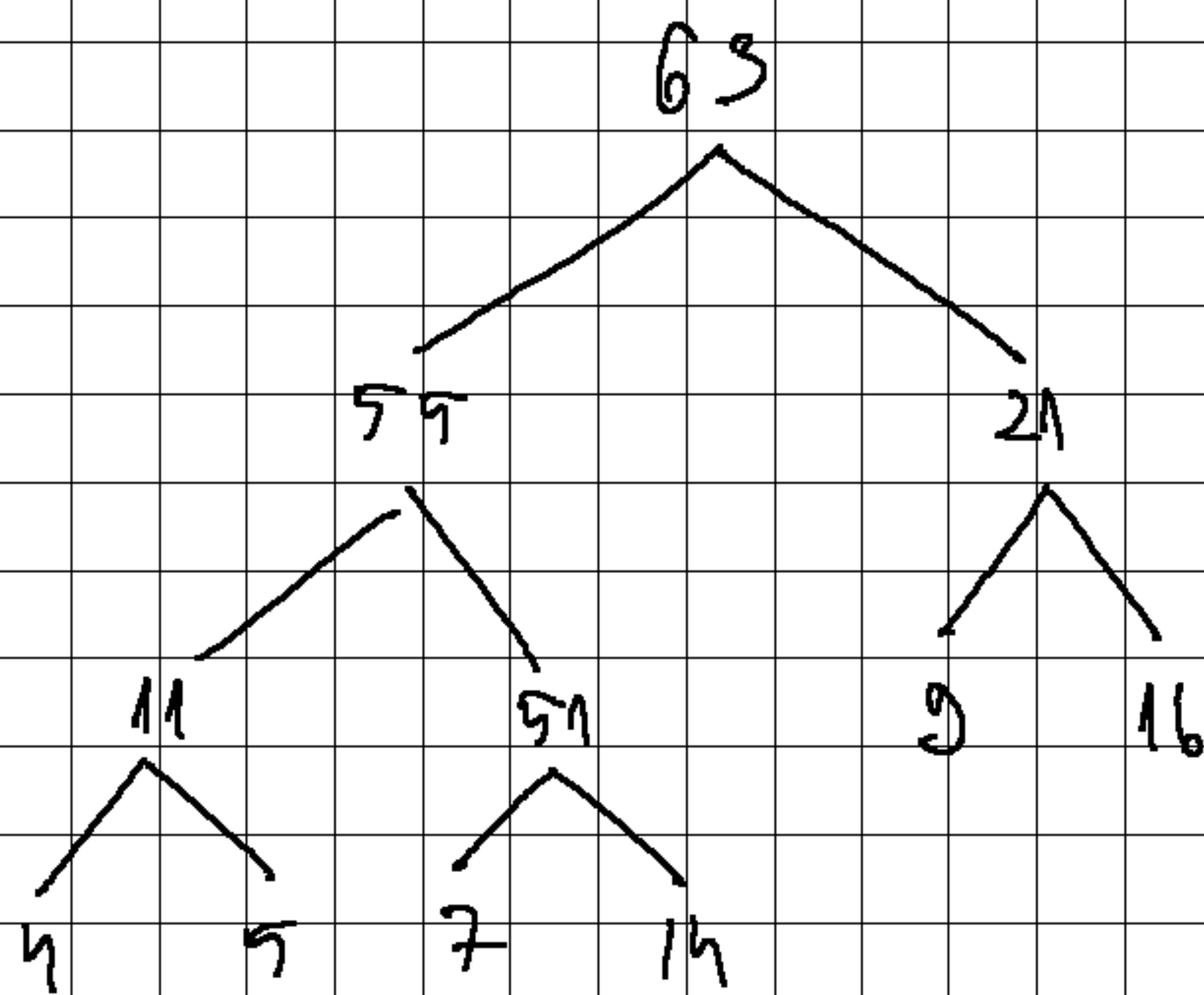
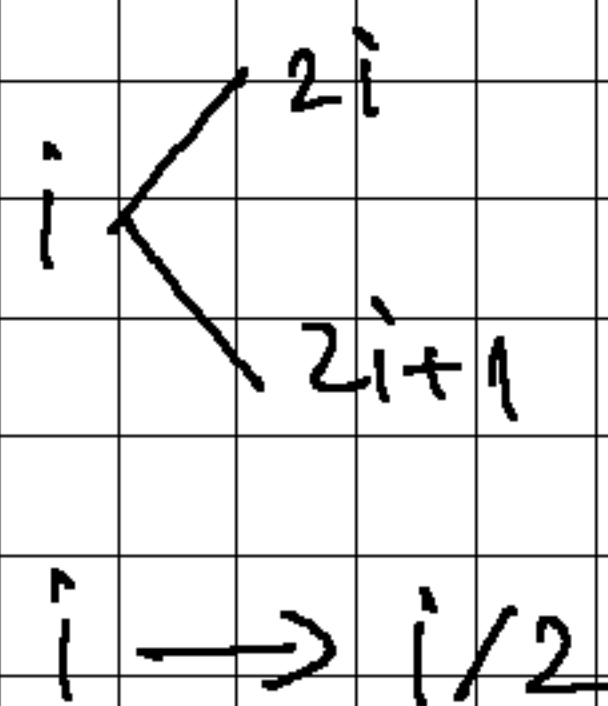


Binary Heap



children



Subalg bubble-up (heap, pos):

currentPos \leftarrow pos

parent \leftarrow currentPos / 2

currentPos > 1 and

while * heap.elements[currentPos] > heap.elements[parent] ex:

tmp \leftarrow heap.elements[currentPos]

heap.elements[currentPos] \leftarrow heap.elements[parent]

heap.elements[parent] \leftarrow tmp

currentPos \leftarrow parent

parent \leftarrow currentPos / 2

$O(\log_2 n)$

Subalg bubble-down (heap, pos):

while pos < heap.len ex:

if $2 \cdot \text{pos} \leq \text{heap.len}$ then

maxchild $\leftarrow 2 \cdot \text{pos}$

if $2 \cdot \text{pos} + 1 \leq \text{heap.len}$ and

$\text{heap.elems}[2 \cdot \text{pos} + 1] > \text{heap.elems}[\text{pos}]$ then

maxchild $\leftarrow 2 \cdot \text{pos} + 1$

if $\text{heap.elems}[\text{maxchild}] > \text{heap.elems}[\text{pos}]$ then

tmp $\leftarrow \text{heap.elems}[\text{maxchild}]$

$\text{heap.elems}[\text{maxchild}] \leftarrow \text{heap.elems}[\text{pos}]$

$\text{heap.elems}[\text{pos}] \leftarrow \text{tmp}$

pos $\leftarrow \text{maxchild}$

else

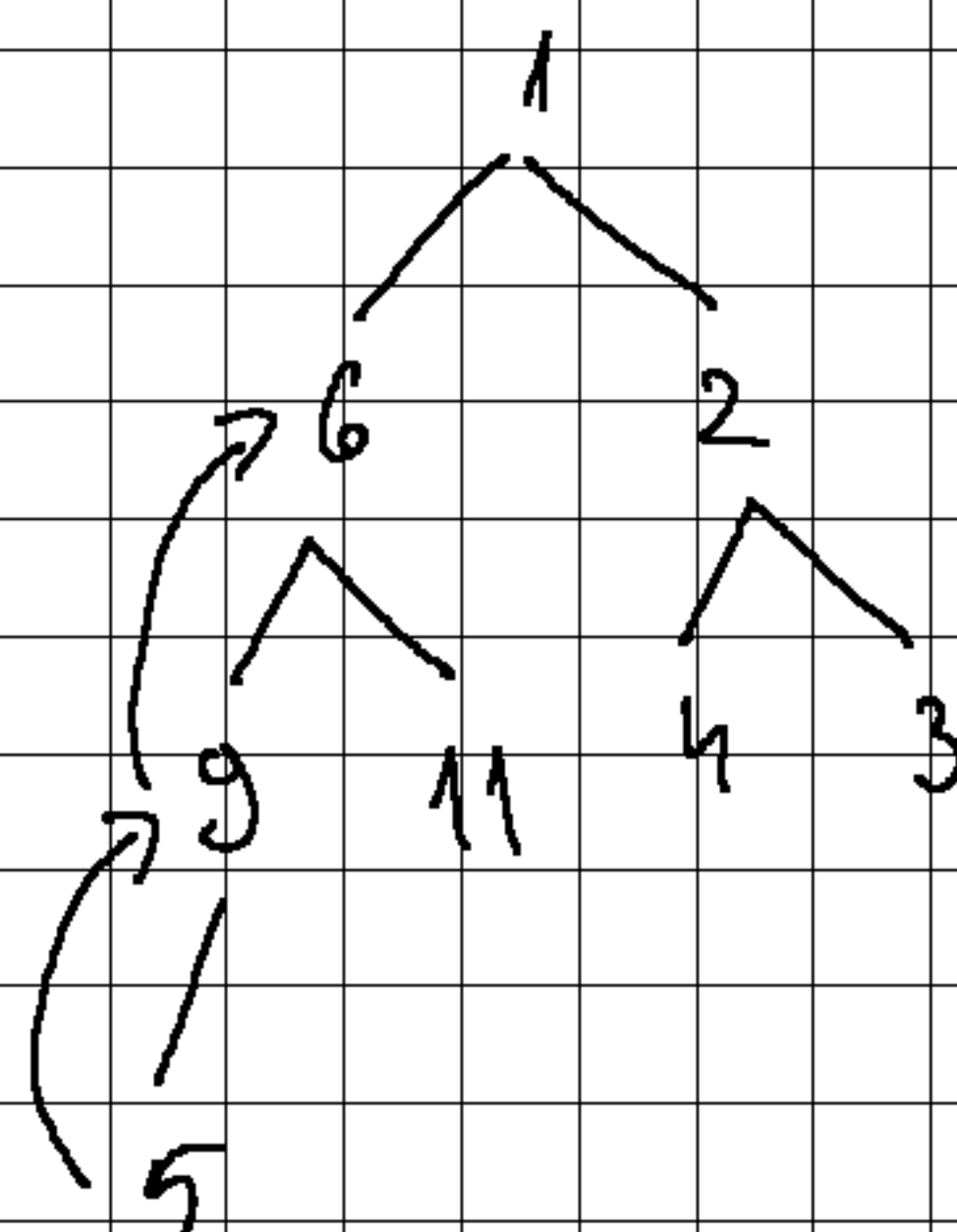
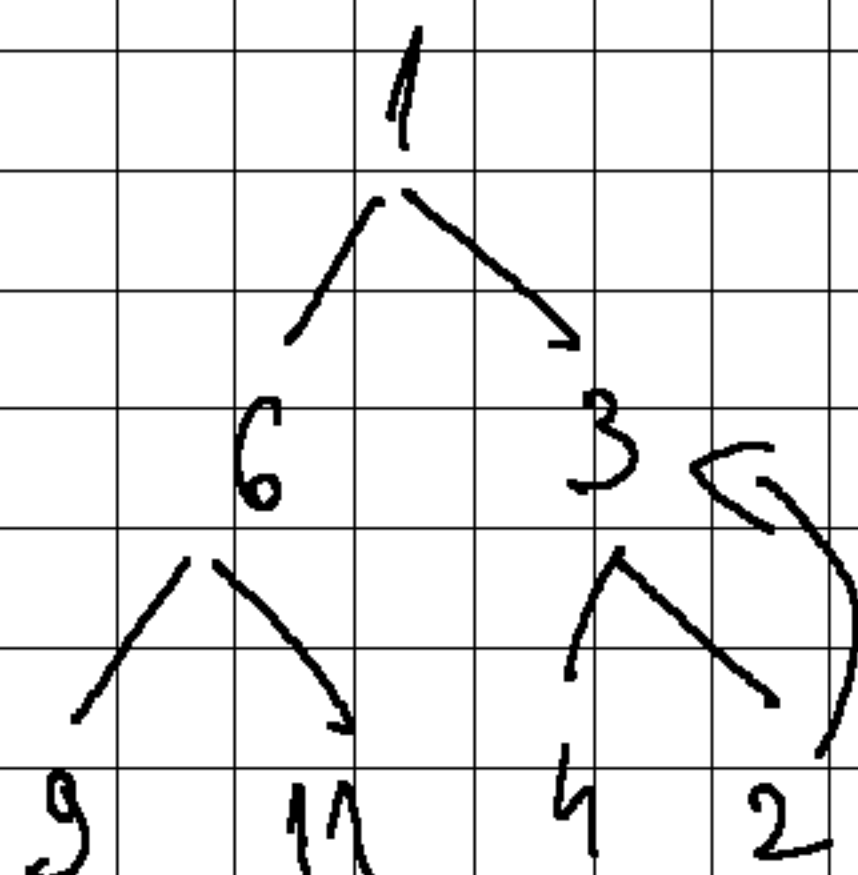
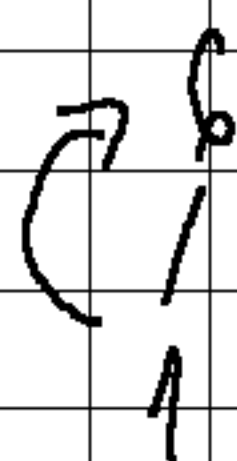
pos $\leftarrow \text{heap.len}$

else

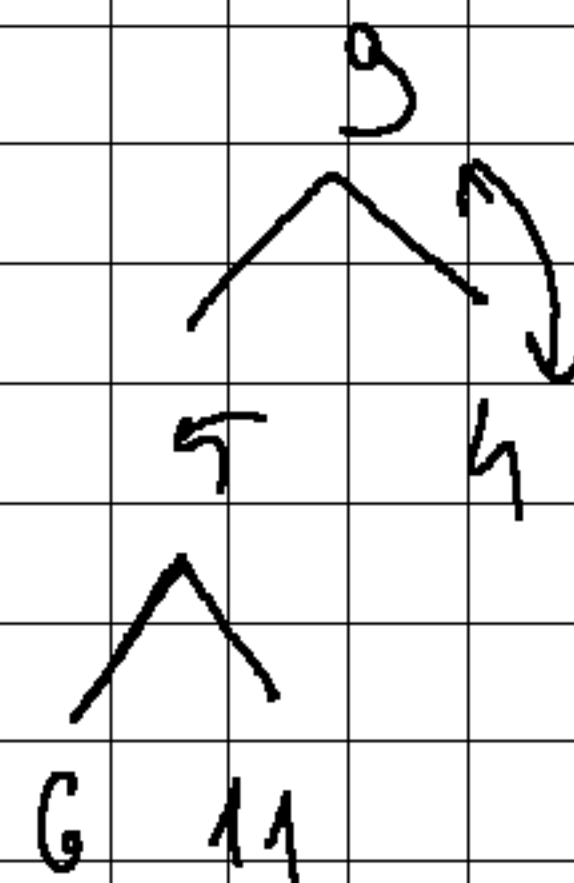
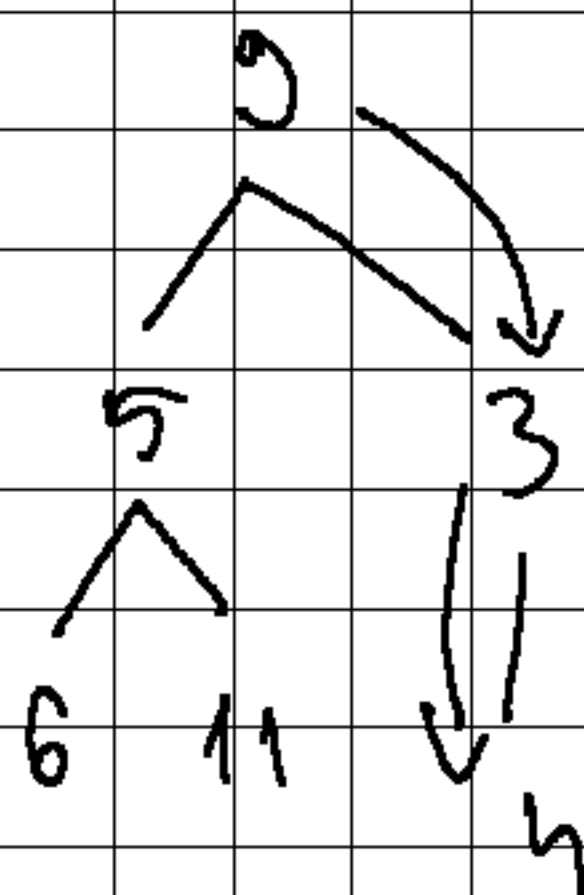
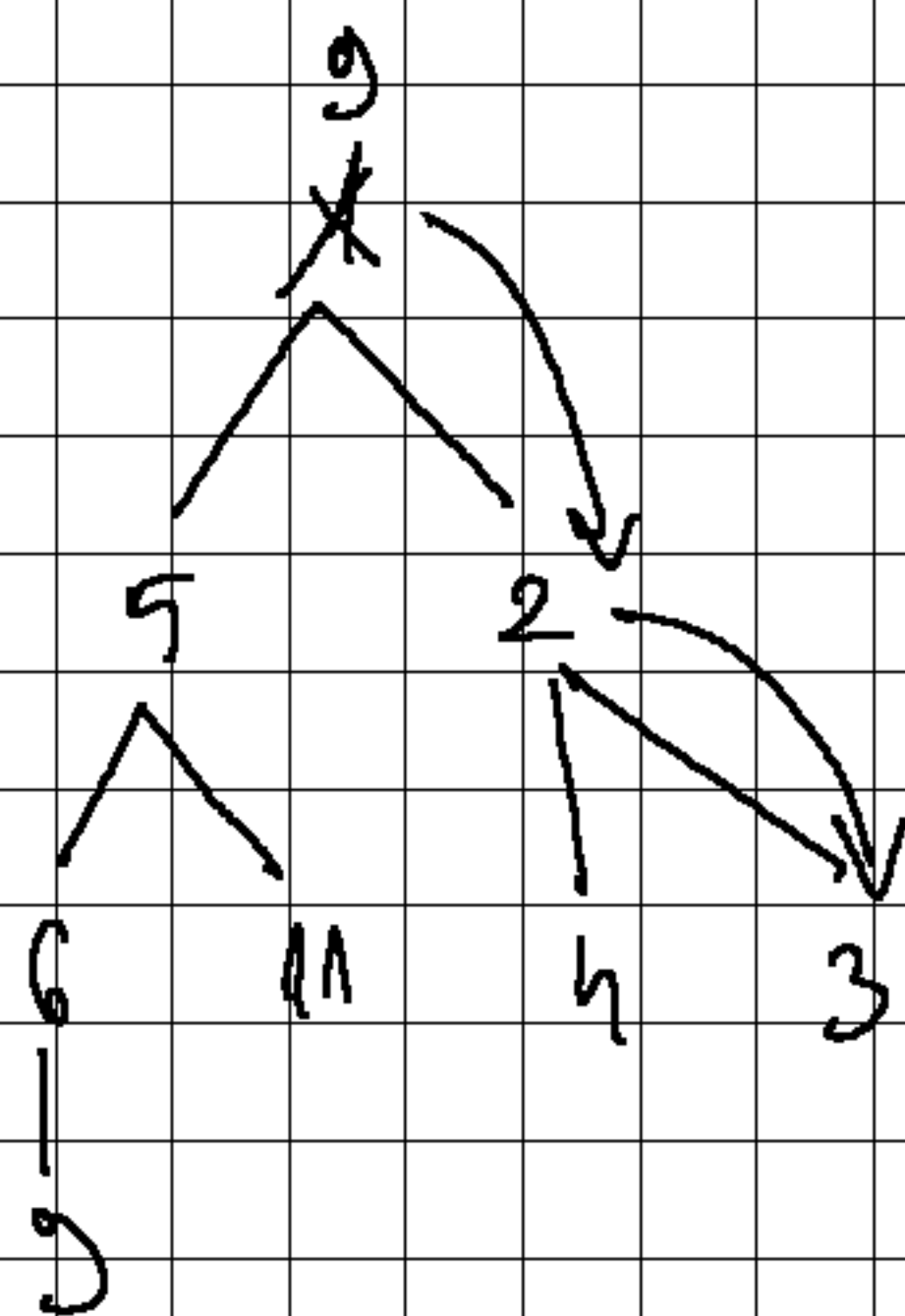
pos $\leftarrow \text{heap.len}$

Heap Sort: $n \log n$

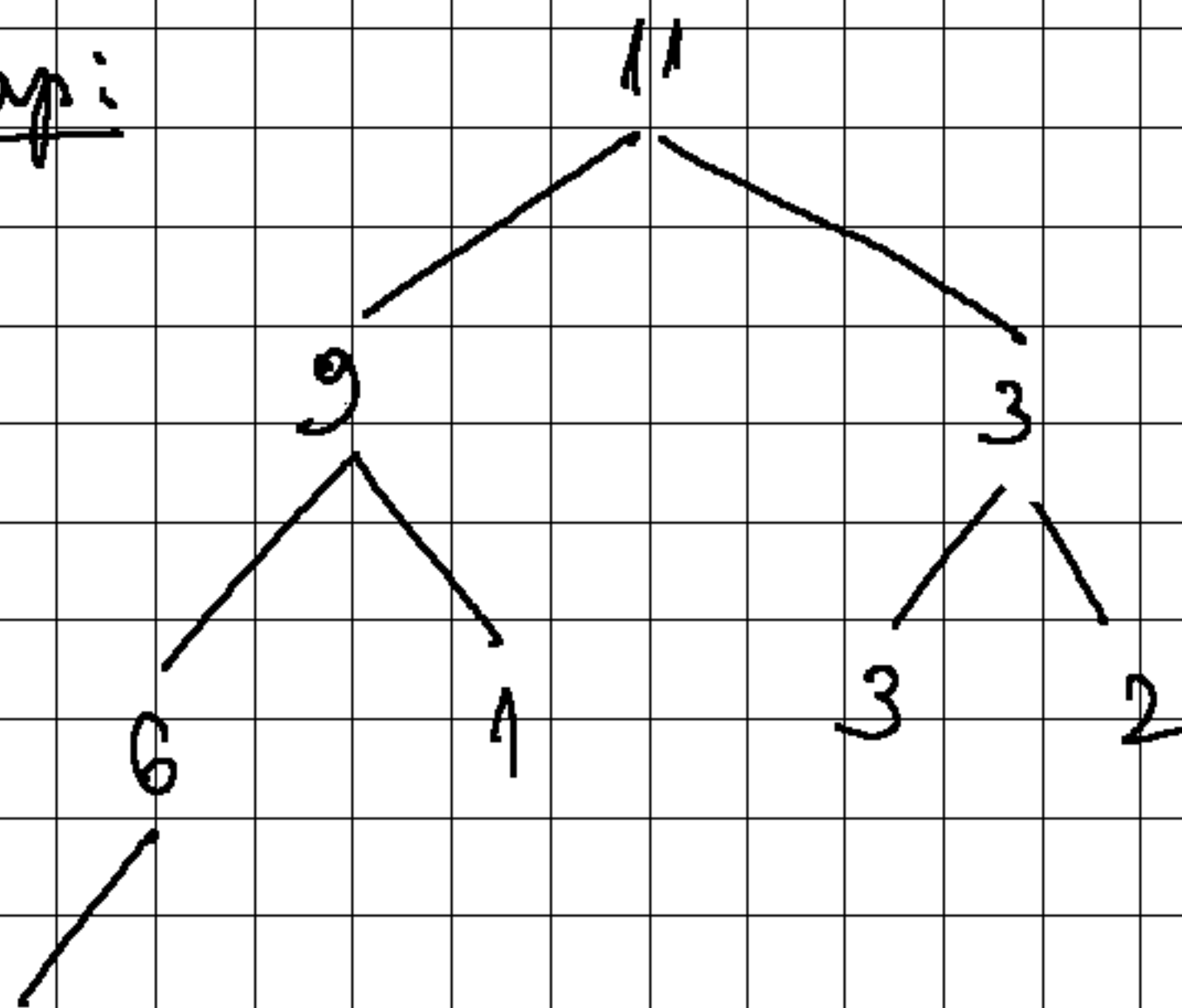
6 1 3 9 11 4 2 5



Min heap.



Max heap:



$\Theta(m)$

$m/2 \rightarrow 0$

$m/4 \rightarrow 1$

$m/8 \rightarrow 2$

\vdots

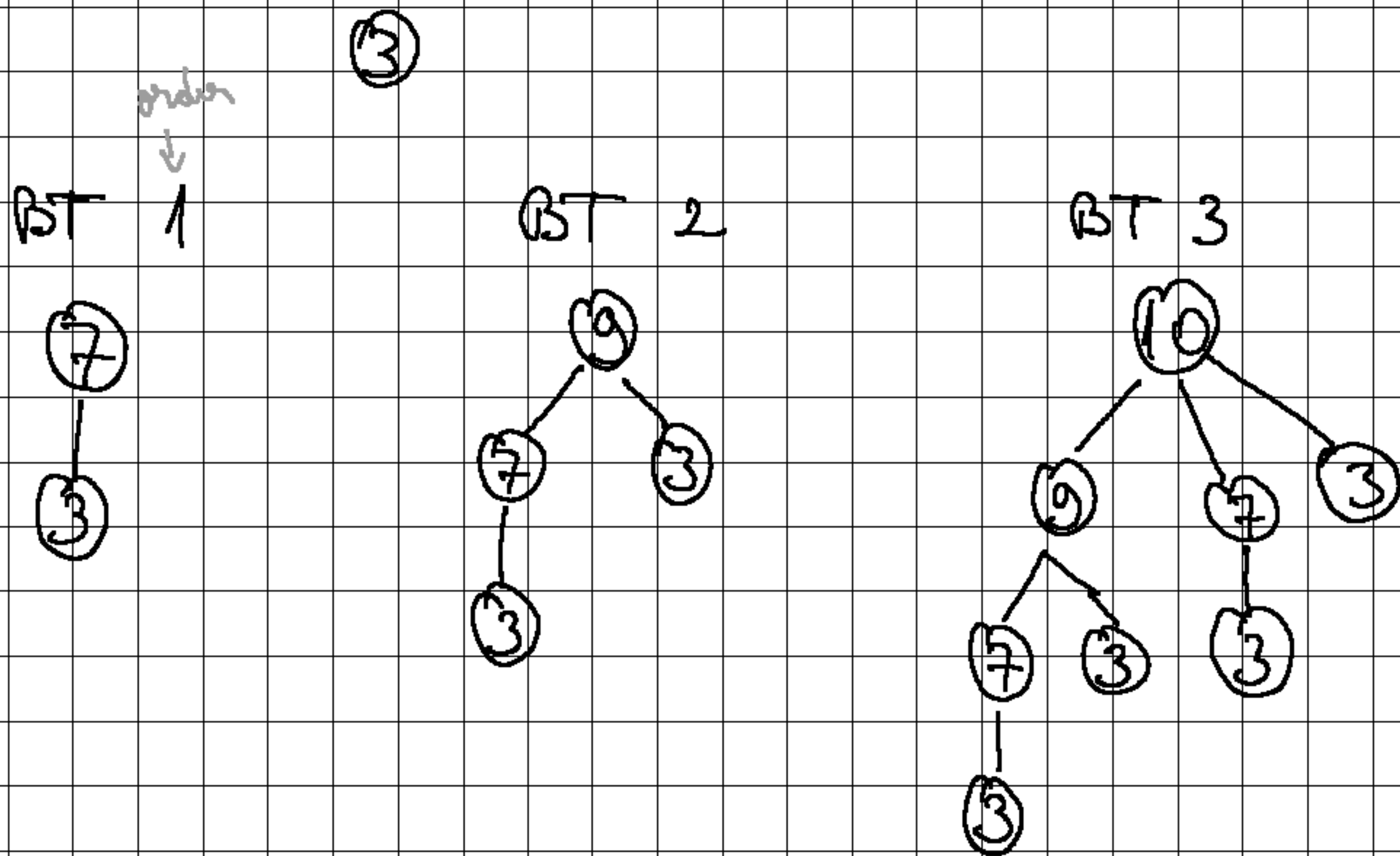
1

\vdots

$\rightarrow \log m$

Binomial heap:

Binomial tree: order 0



Binomial tree of order k has 2^k nodes

Height of BT of order k is k

Merging:

