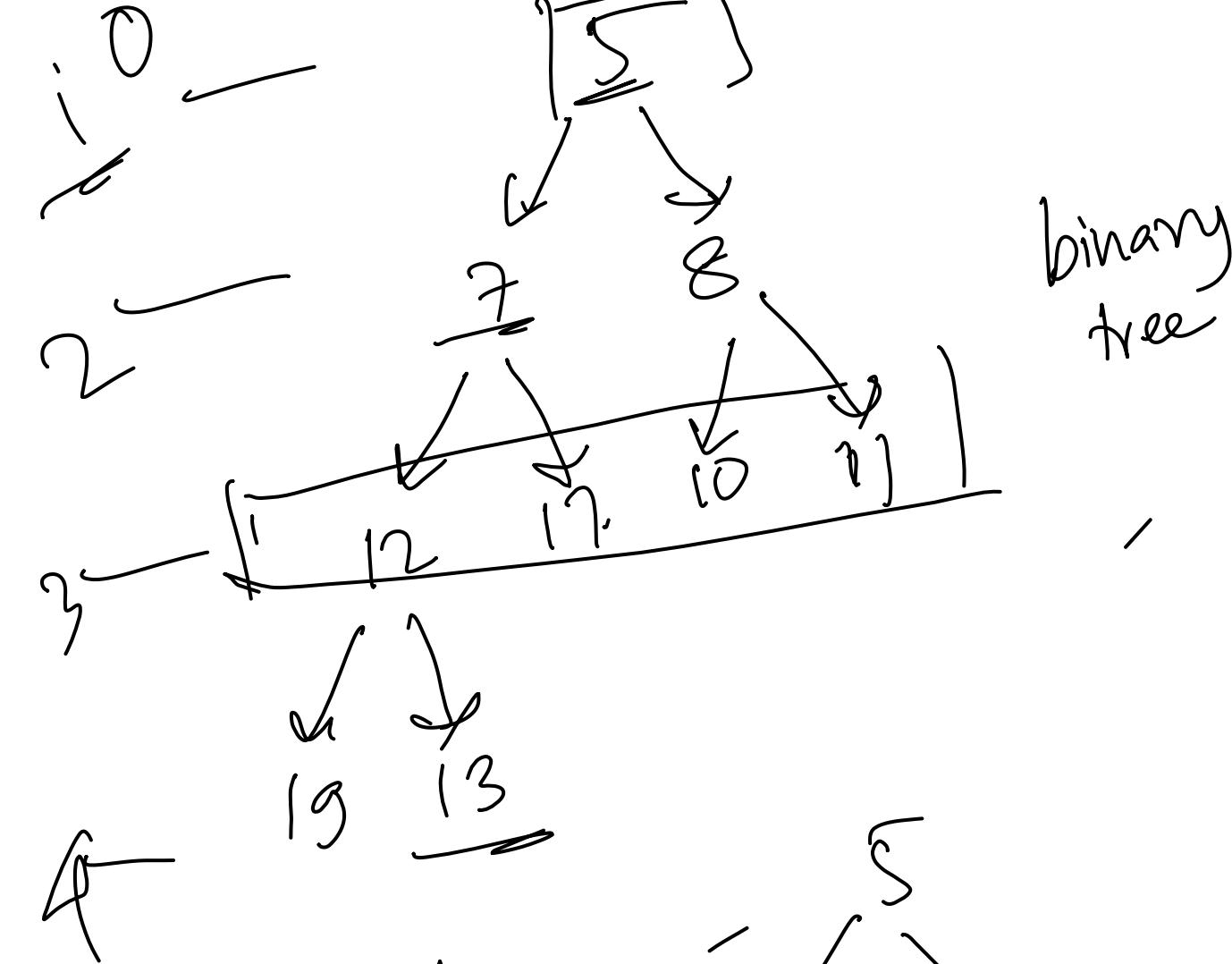
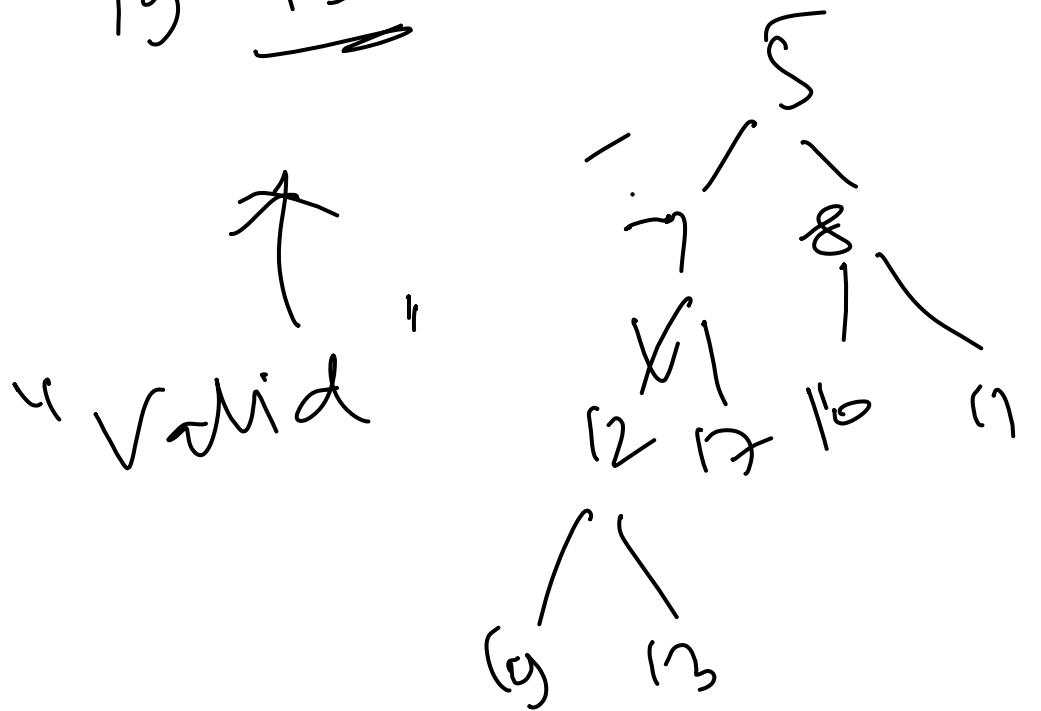


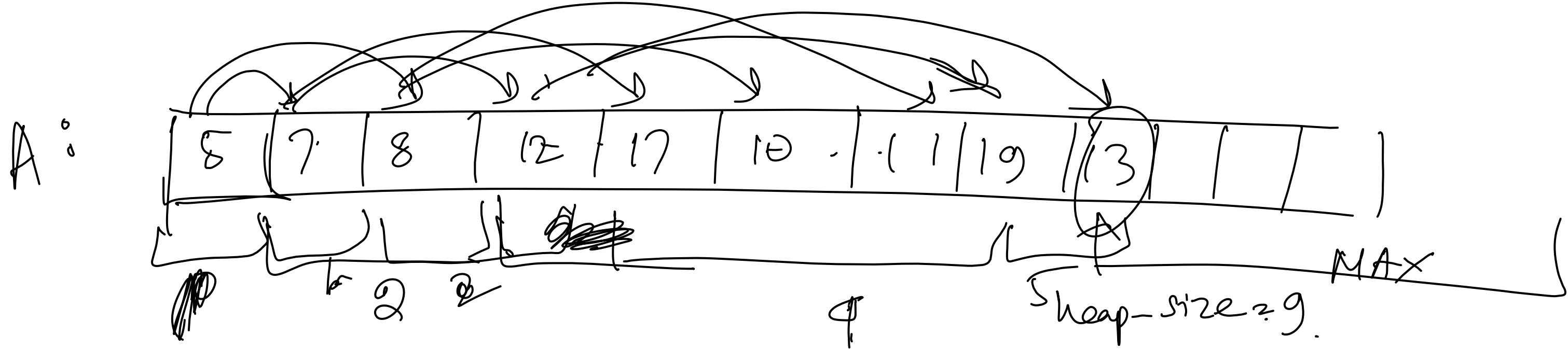
Binary (min) Heaps



- each vertex is a key.
 - every parent has 2 children & every child node has got a unique parent.
 - All but last level is full
 - last level is full left to right
(no gaps)
- only the last level's parent can have no children as long as it satisfies



- Parent's key is smaller than that of the children.



Procedure Left(x)
 // left returns the ^{index of the} left child of the node at index x.
 Return $2x$.

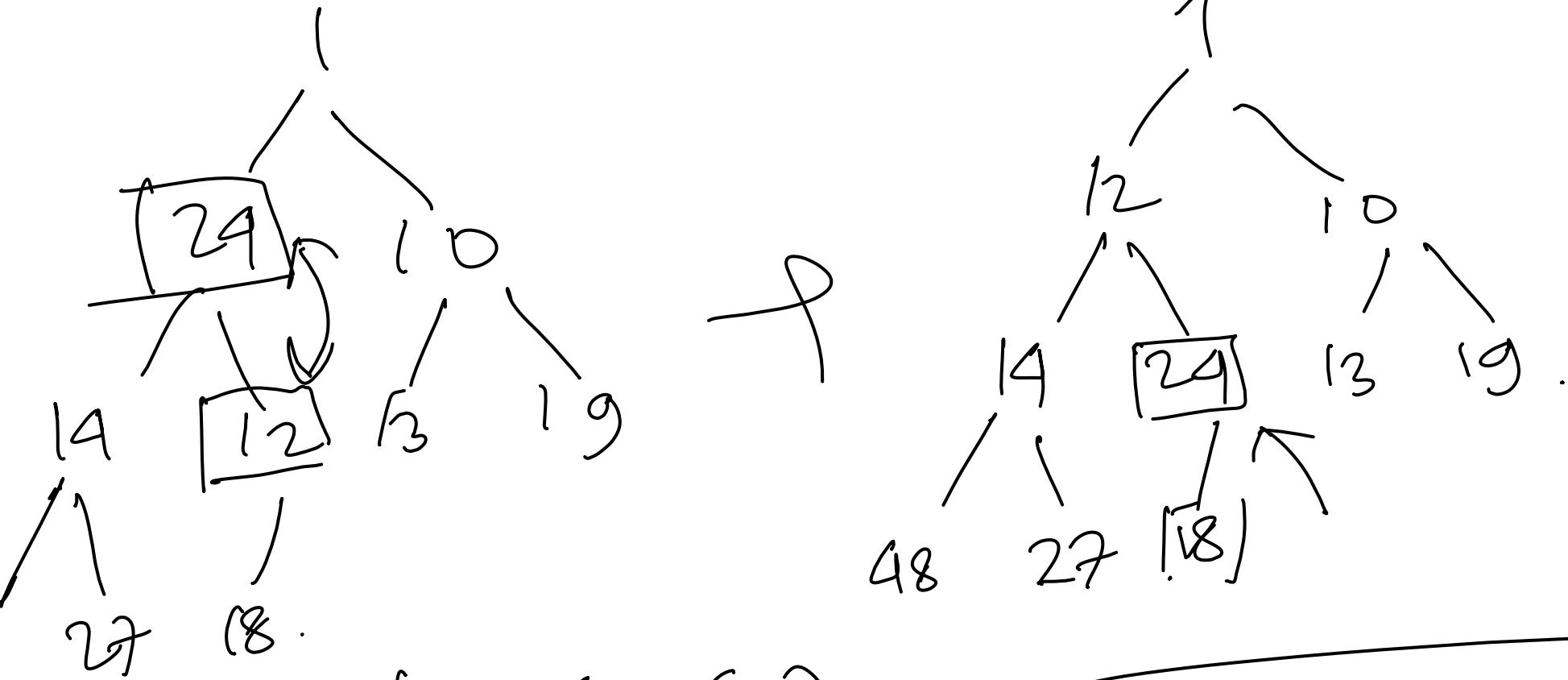
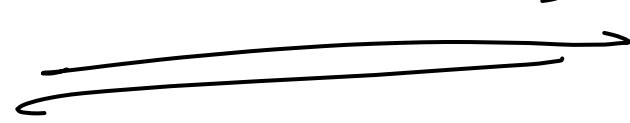
Procedure Right(x)

Procedure Parent(x)

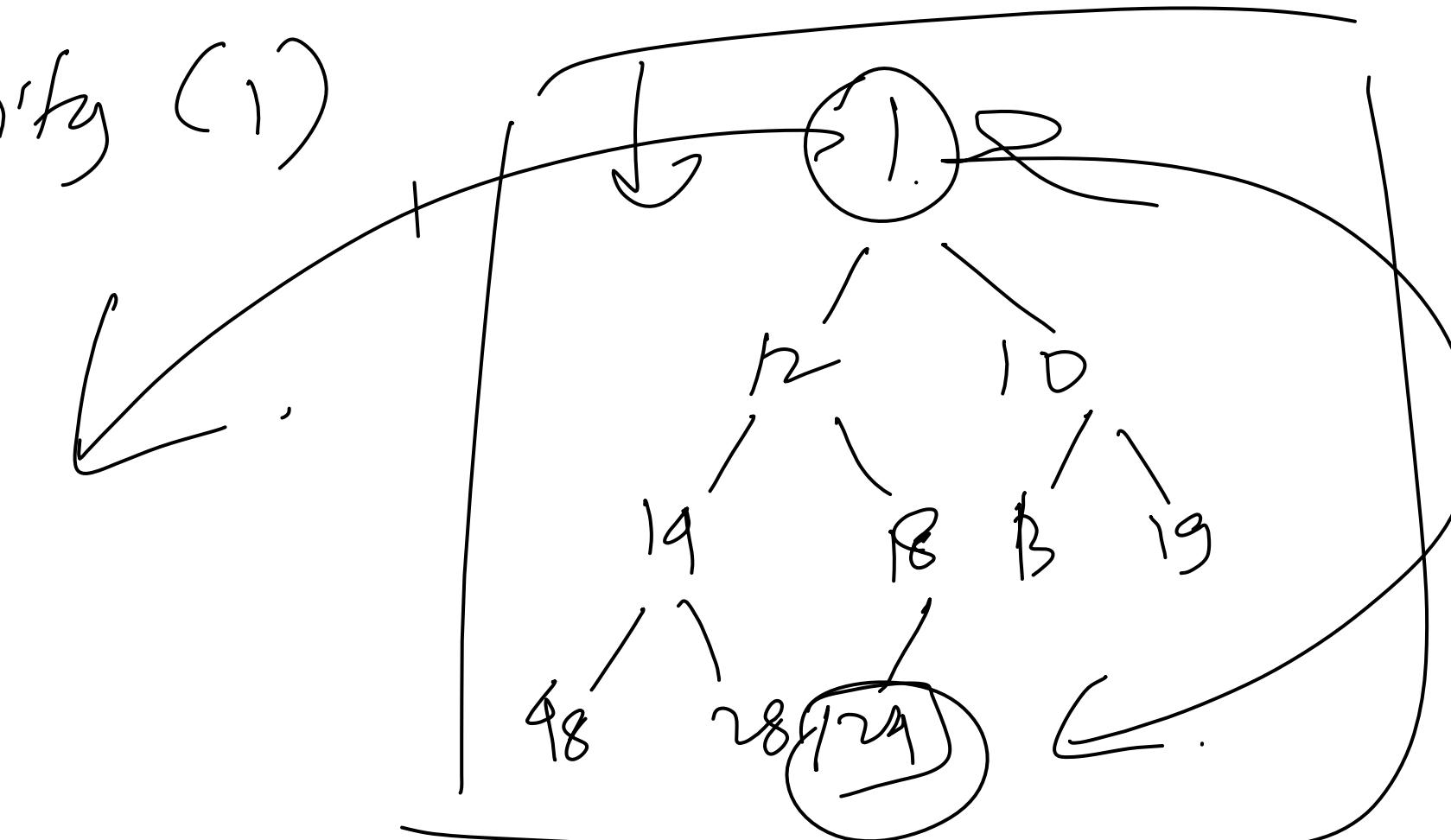
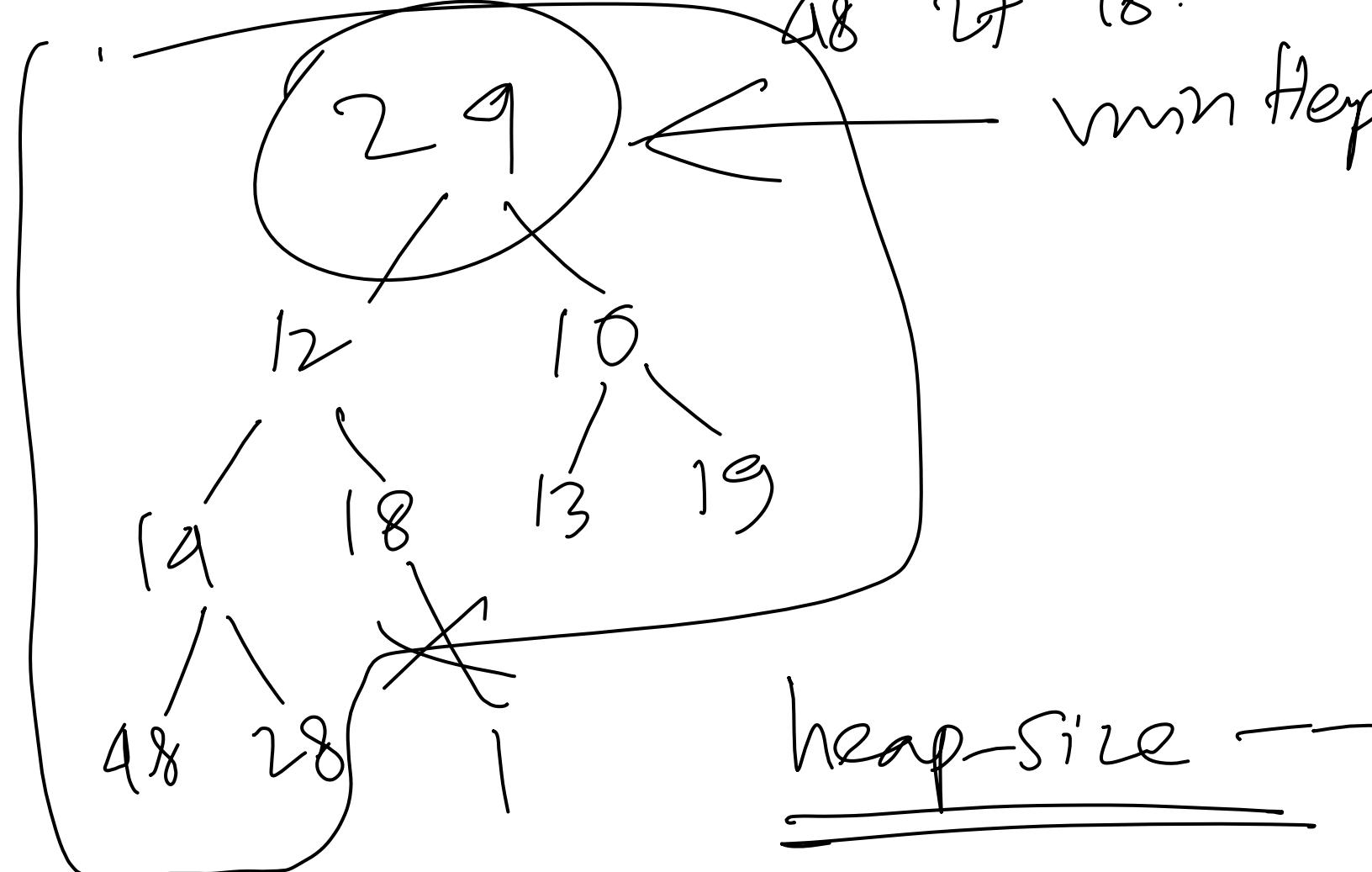
Return $\frac{x}{2}$

) Struct
key

Min-heapify



min-heapify (1)



Procedure Min-Heapify (A, x)

// A has a violation of heap property at index x .

while ()

{

minIndex $\leftarrow x$

if ($\text{left}(x) \leq \text{heap_size} \wedge A[\text{left}(x)].\text{key} < A[x].\text{key}$)

 minIndex $\leftarrow \text{left}[x]$,

else if

 if ($\text{right}(x) \leq \text{heap_size} \wedge A[\text{minIndex}].\text{key} < A[\text{right}(x)].\text{key}$)

~~A[$\text{right}(x)$].key~~ ✓

 minIndex $\leftarrow \text{right}[x]$

endif

if ($\min\text{Index} \neq x$)
~~A~~ temp $\leftarrow A[x]$
 $A[x] \leftarrow A[\min\text{Index}]$
 $A[\min\text{Index}] \leftarrow \text{temp}$.
Min-Heapify ($A, \min\text{Index}$)

else
return
end if
end while.

$$T(l) = T(l-1) + \Theta(1) = \Theta(l)$$

log n

Procedure Build-Heap(A, n)

heap-size $\leftarrow n$.

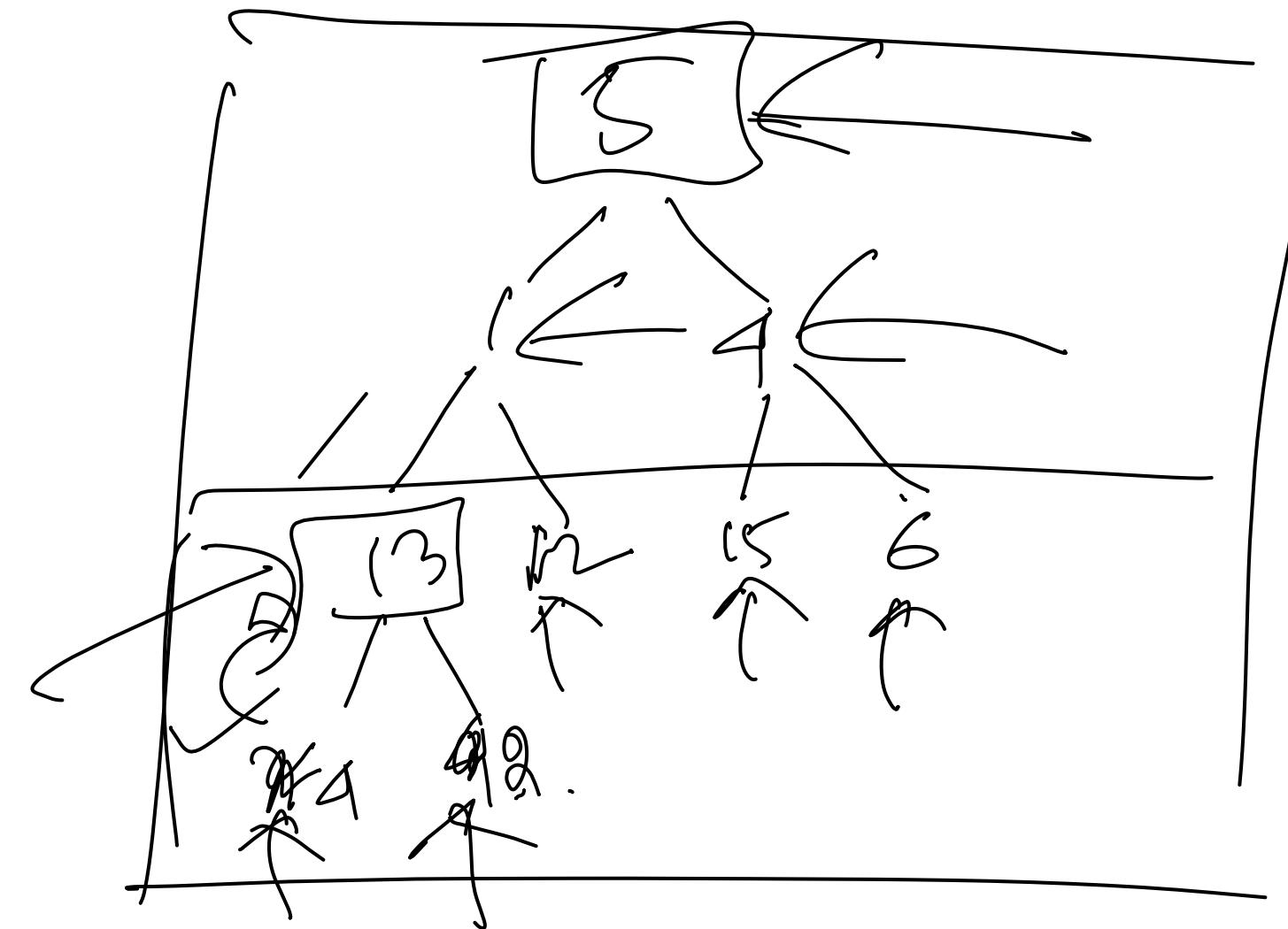
for $j = n$ to ~~to~~ 1

MinHeapify(A, j).

Time Complexity? $O(n \log n)$

$O(n)$

5	1	4	(3/12	15	6	2	9
---	---	---	-------	----	---	---	---



Time Complexity

Q: If there are n items in the heap, what is the height?

R: Suppose height = $\ell \Rightarrow (\ell-1)$ levels are full.

levels : $\begin{matrix} 1 & + 2 & + & 2^2 \\ \uparrow & \uparrow P & & \\ 0 & 1 & 2 & \end{matrix}$... $+ 2^{\ell-1} + \underbrace{[1, 2^\ell]}_{\downarrow \text{last levels.}}$

$= 2^\ell - 1 + [1, 2^\ell]$

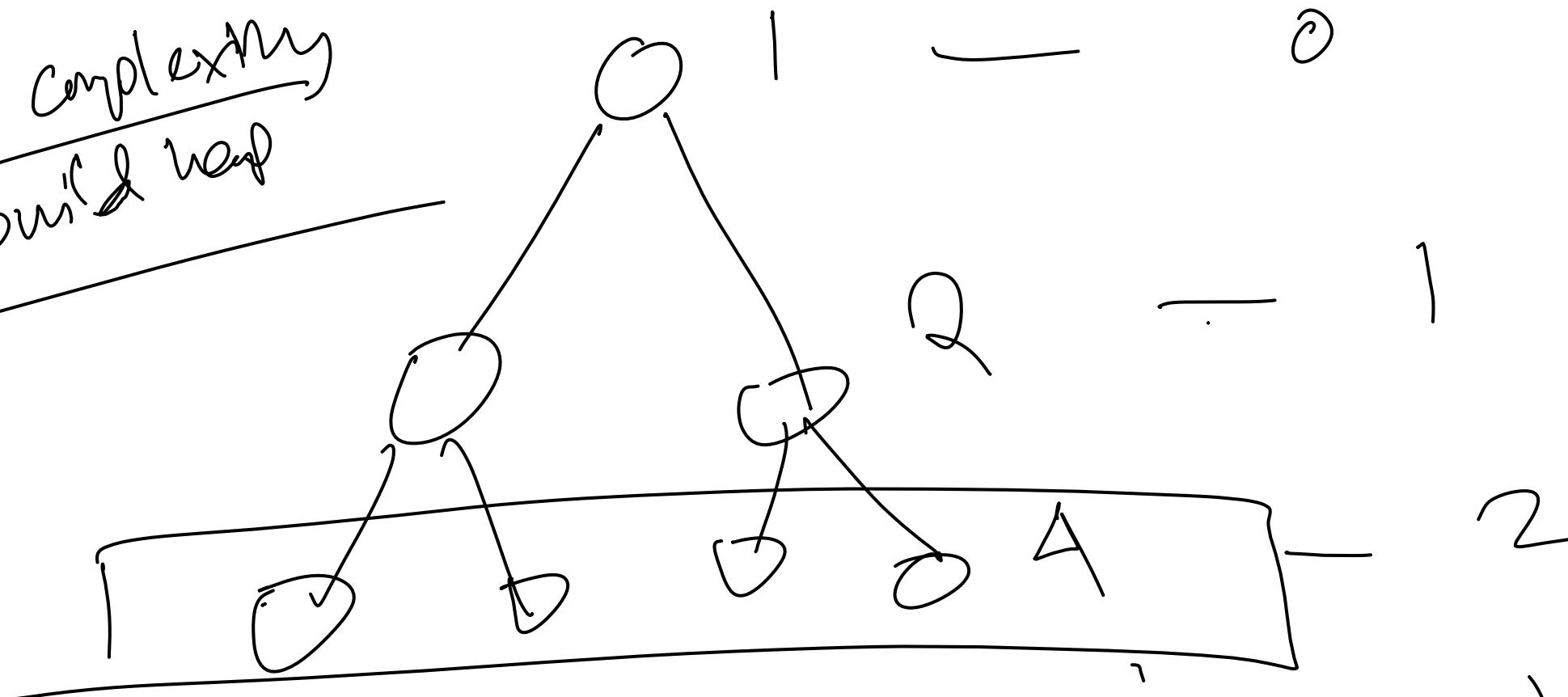
$\boxed{[2^\ell, 2^{\ell+1}-1]}$

$\ell = \Theta(\log n)$

Time complexity of Min-Heapify:

$O(\log n)$.

Time complexity
of Build heap



$\theta(\log n)$

Time
 $\theta(l-j)$

Time
 $\theta(l-j)$



$\theta(n)$.
 $\theta(n)$.

$\theta(n)$

$$C \cdot \sum_{j=1}^{\log n} 2^j (Q - \underline{j}) = \Theta(n)$$

$|T|_1 - |T|$

Procedure Extract Min (A)

```

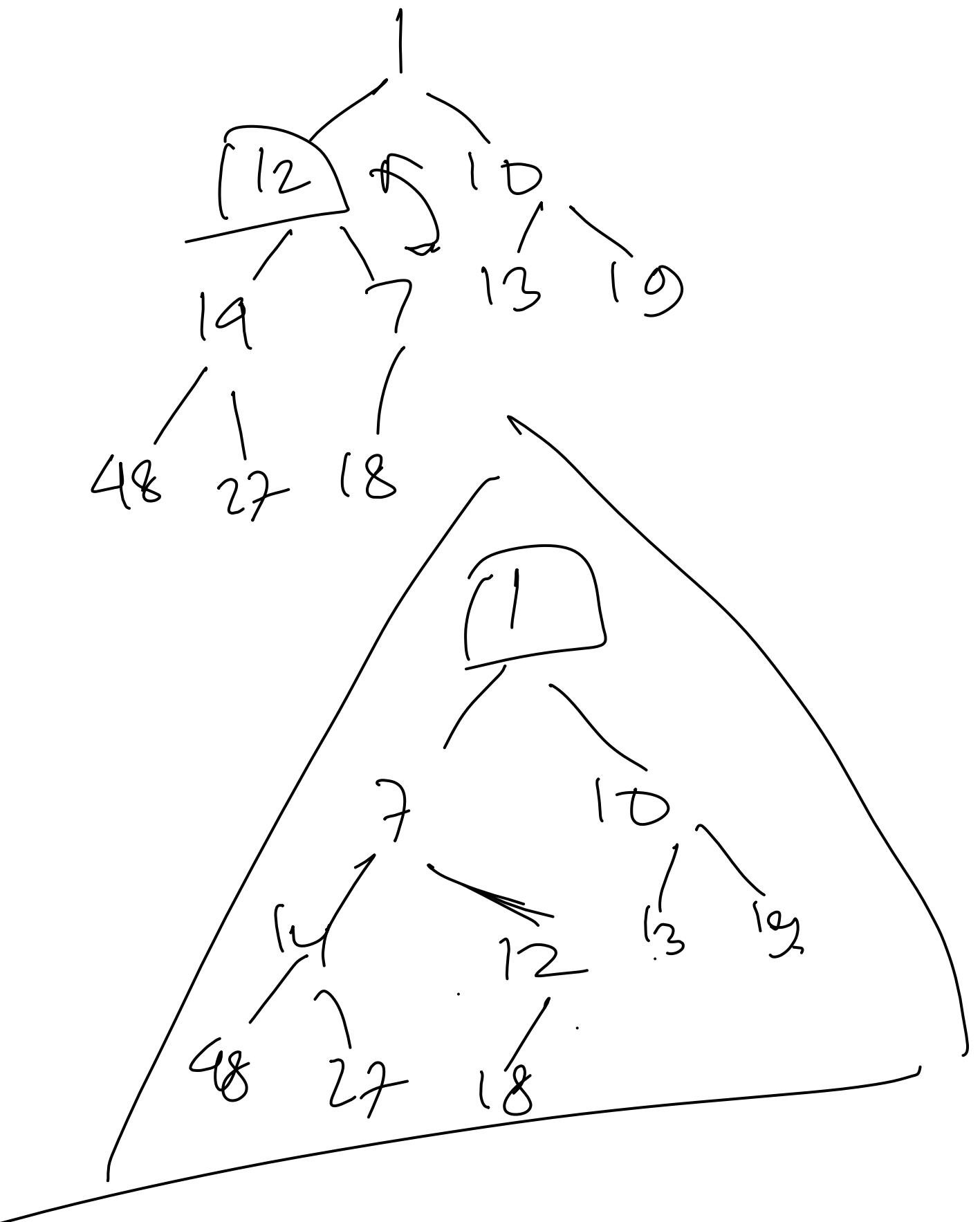
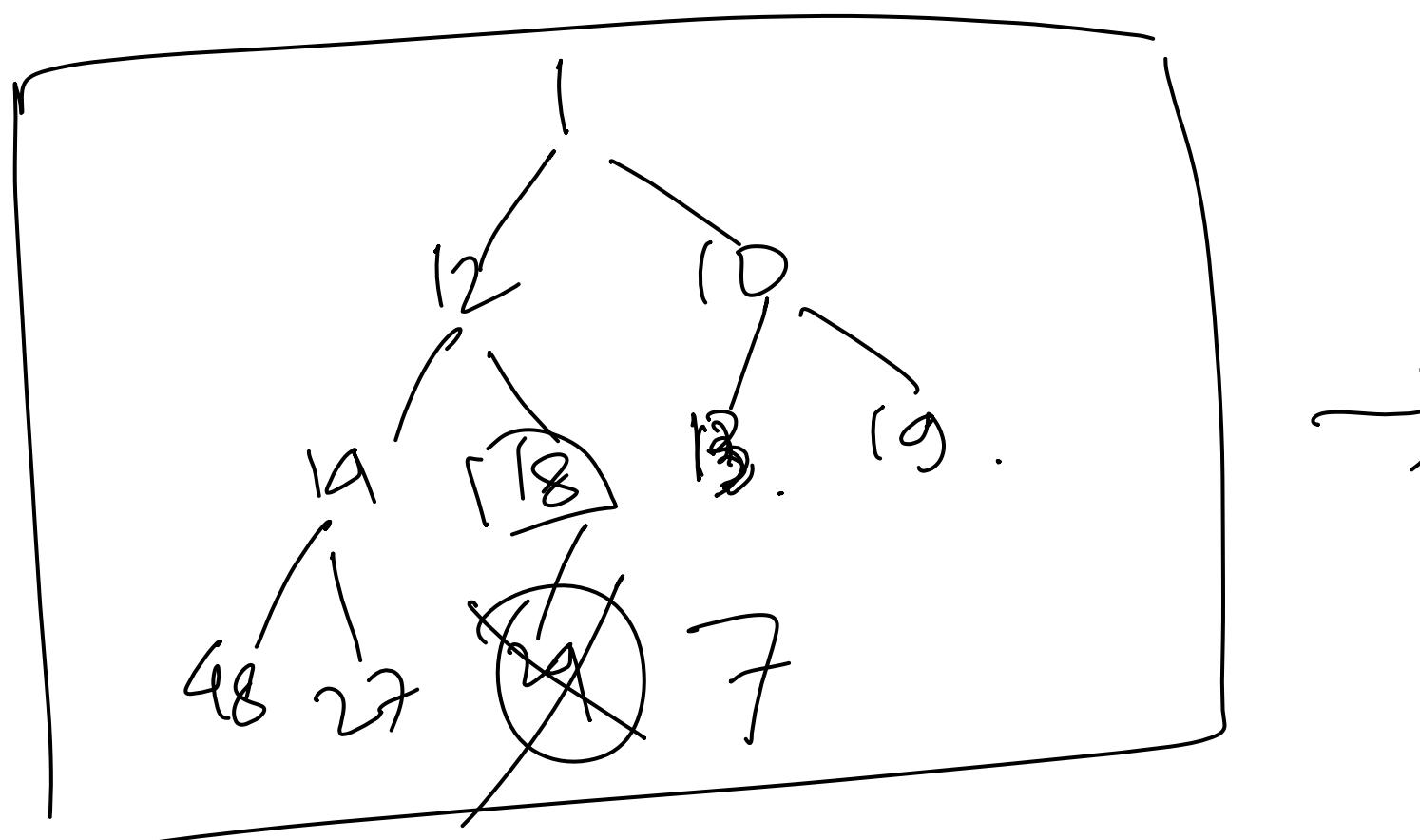
f. temp <= A[1]
A[1] <= A[heap-size]
A[heap-size] <= temp
{ heap-size --
```

MinHeapify ($A, 1$)

$\rightarrow O(\log n)$

Procedure Heap Sort ('A_n)
B ← Build Heap (A, n).
heap-size ← n.
for j = 1 to n
Print('Extract Min (B)')

Correctness ✓
Time Complexity
O(n log n)



Procedure Decrease-Key (A, X, newkey)

if (X > heap-size) nehm emr.

A[X].key ← newkey .

while (1)

{
if (parent(x) = 0) break .

~~loop~~
if (A[parent(x)].key < A[x].key) break

temp < A[x]

A[x] < A[parent(x)]

A[parent(x)] < temp .

```
{  
    x ← parent(x).  
    el while  
    }.
```

$O(\log n)$ operations.

28.08.2024

Priority Queue

Key
data :-

- extract-min : returns the item with smallest key & removes the item from PQ.
- d(v) : stores a vertex \rightarrow key mapping and when queried, returns the key of the queried vertex
- decrease-key : If reduces the key of an item to a new value -