1) BST

RED-BLACK TREES

- NODES ARE RED / BLACK
- ROOT IS BLACK
- LEAVES ARE BLACK
  - NIL
- NODE ~~BLACK~~ RED
  - PARENT BLACK

AT MOST HALF OF EACH PATH CAN BE RED

- FOR EACH NODE x
  ALL PATHS ~~TO~~ x
  FROM x TO A LEAF
  SAME ~~BLAC~~ # BLACK NODES

$\Theta(h)$

HEIGHT

- INSERT
- DELETE
- FIND(k)
- NEXT, PREV
- MAX, MIN

$\Theta(n)$

NO

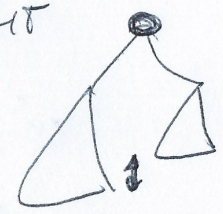
3 BLACK

4 BLACK

$\log(n)$

≈ BALANCED

- HEIGHT



- WEIGHT

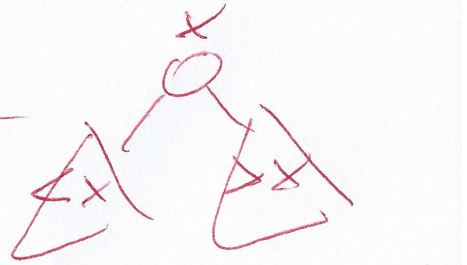
(2) RED BLACK TREES

$\longrightarrow$ height $\leq 2\lceil \log_2 n \rceil \simeq \Theta(\log n)$

ROTATIONS $\Theta(1)$

(1) TRANSFORM ANY BST IN ANY BST TIME

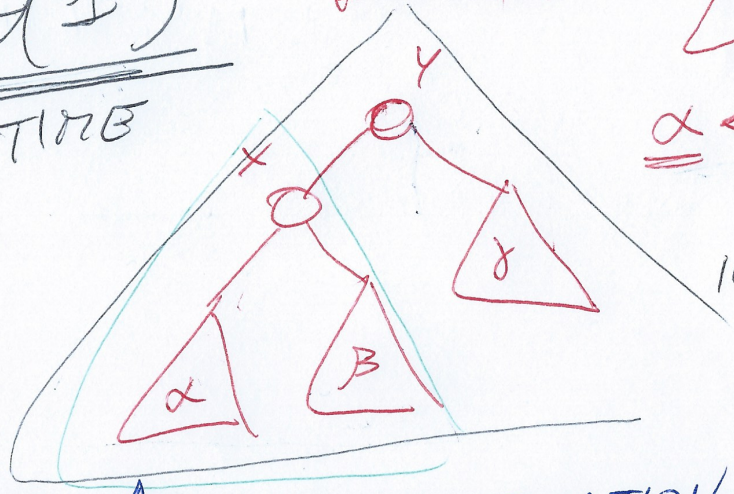
BST



$\alpha < x < \beta < y < \gamma$

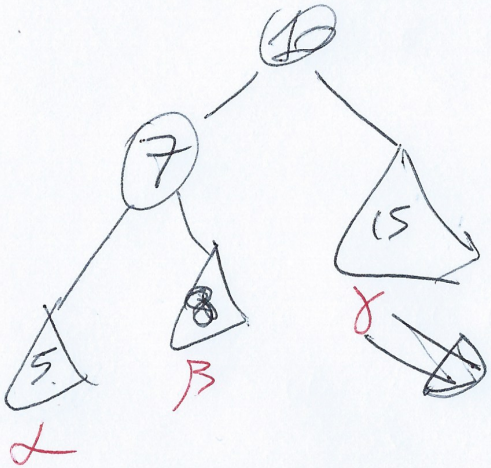
IN-ORDER VISIT

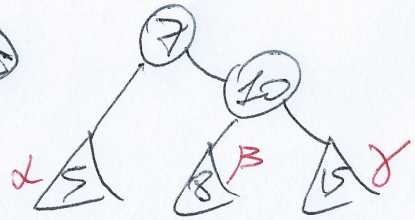
SORTED SEQUENCE

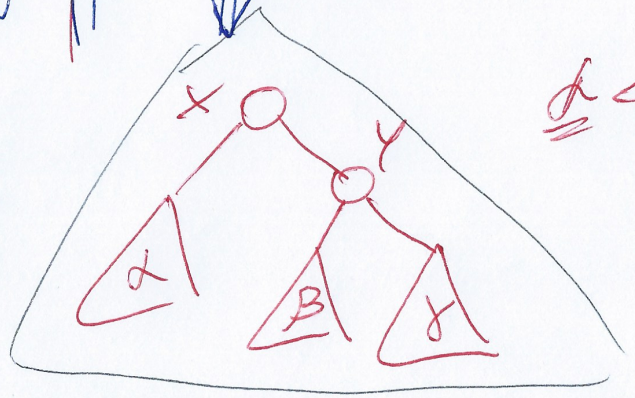


LEFT ROTATION

RIGHT ROTATION







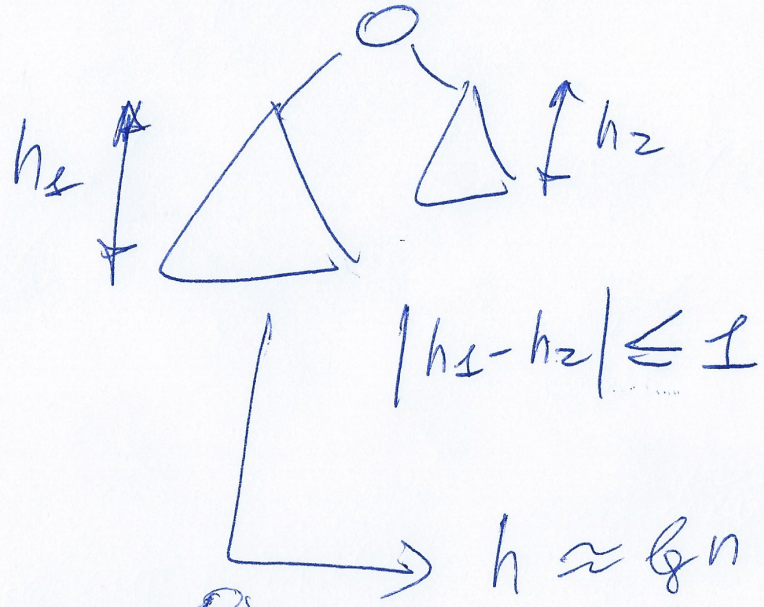
$\alpha < x < \beta < y < \gamma$

③ AVL-TREES

ADELSON-VELSKY, LANDIS

BALANCED IN HEIGHT

FOR EACH NODE



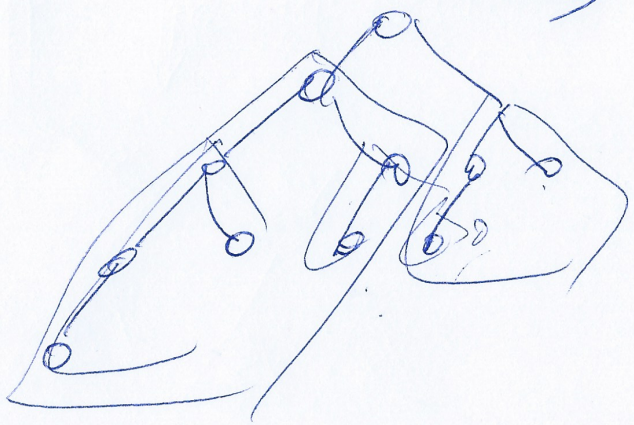
INSERT

DELETE

→ HOW MANY ROTATIONS ?

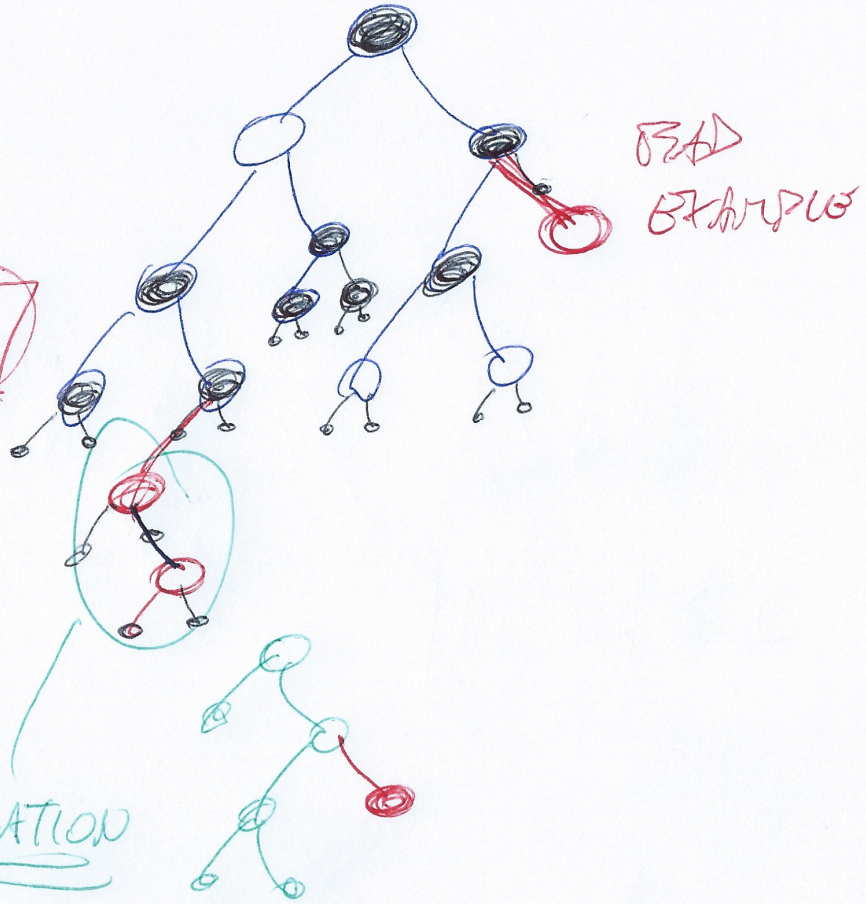
↓

$\log n$ PER OPERATION



(4) INSERTION IN RED BLACK TREES

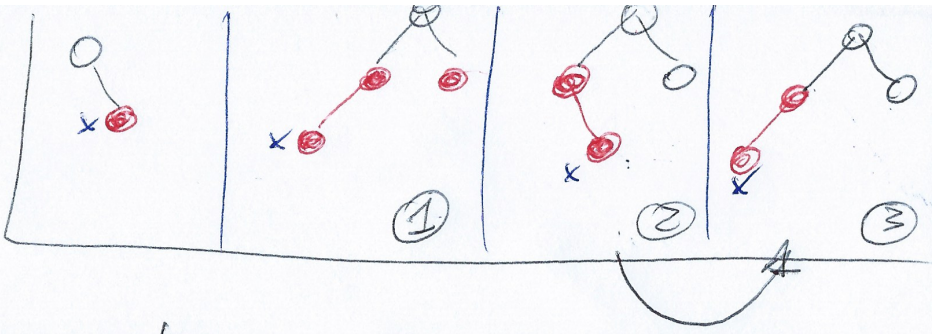
- INSERT AS IN A BINARY SEARCH TREE
  → ALWAYS A LEAF IS RED
  WITH 2 BLACK EMPTY CHILDREN



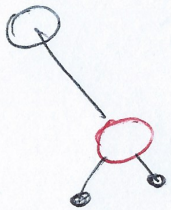
OK W.R.TO. BLACK LENGTH OF PATHS

PROBLEM: IF THE PARENT OF THE INSERTED NODE IS RED

(5) FULFIL THE RB PROPERTY

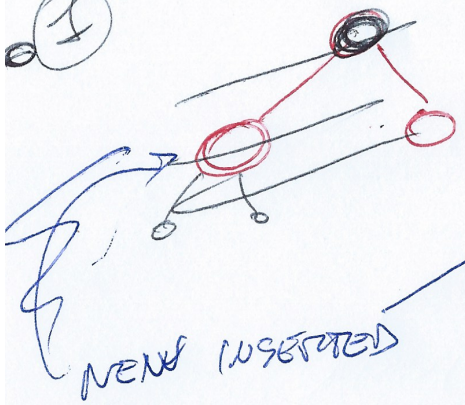


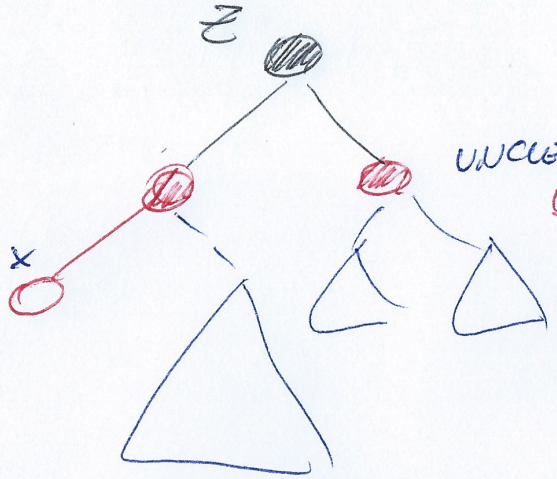
(0)



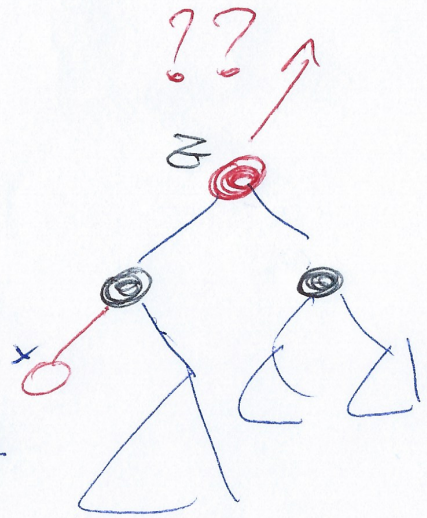
PARENT ALREADY BLACK ⟹ OK

(1)





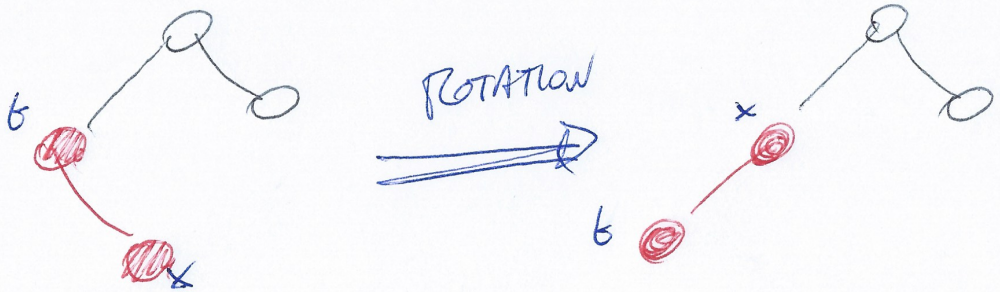
COLOUR TRANSFORM ⟹



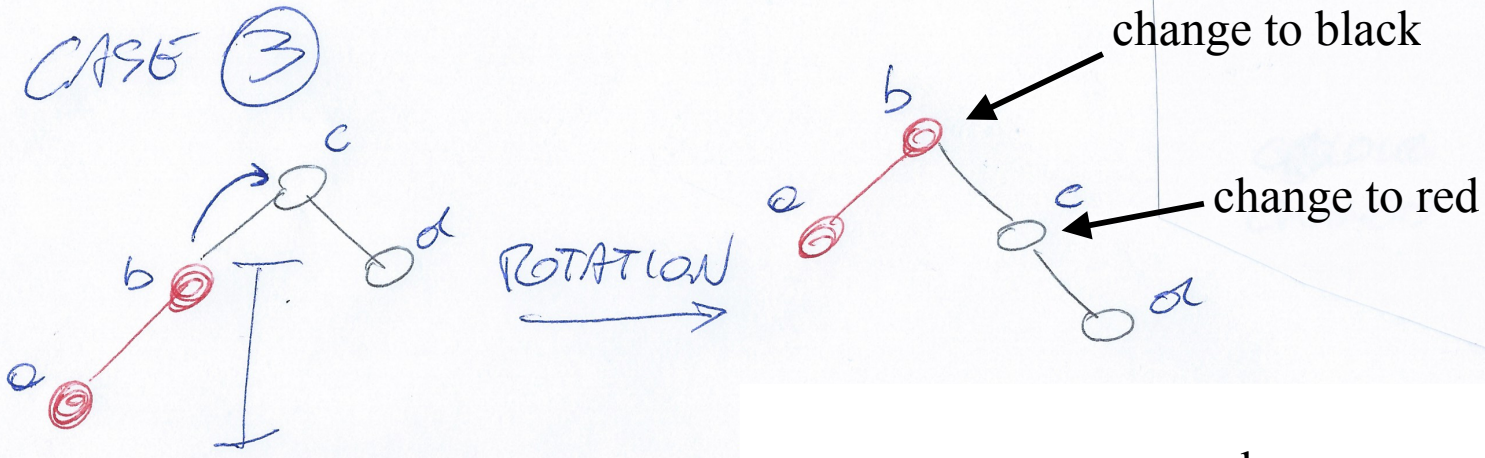
BLACK LENGHT OF ANY PATH TO LEAF IS AS BEFORE

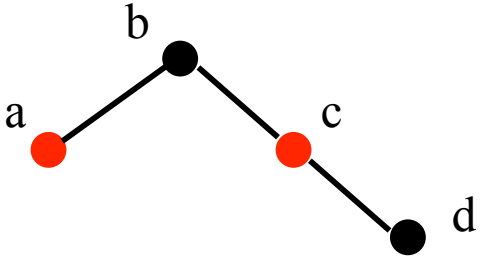
z MIGHT HAVE A RED PARENT ⟹ SAME PROBLEM ONE LEVEL UP

(6) TRANSFORM CASE (2) IN CASE (3)



CASE (3)





7) **PRIORITY QUEUE** (ADS)

- SET OF ELEMENTS ($<$) TOTALLY ORDERED

OPERATIONS:

- INSERT (x)
- TOP (GET THE MAXIMUM ELEMENT)
- POP (REMOVE THE MAXIMUM)
- INCREASE KEY (x, y)

SOLVE BY RED BLACK TREES

$\Theta(\lg n)$ PER OPERATION

SORTING N ELEMENTS

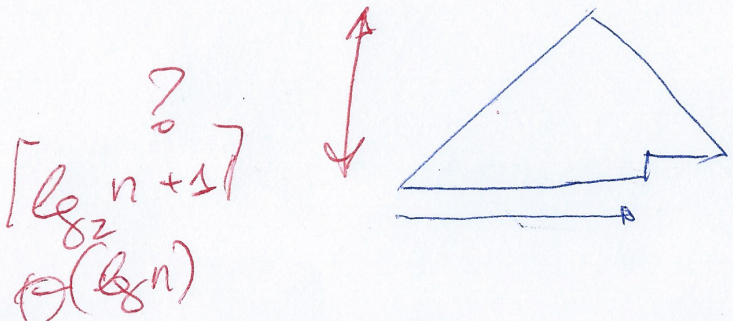
N INSERTION + N MAXIMUM OP'S

$\Theta(n \lg n)$ SORTING

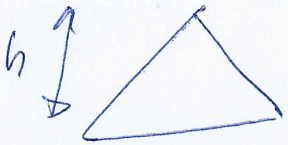
(8) HEAP ⟶ (HEAPSORT ALGOR)

(DISTINCT ELEMENTS)

- COMPLETE (≈) BINARY TREE ~~SEARCH~~

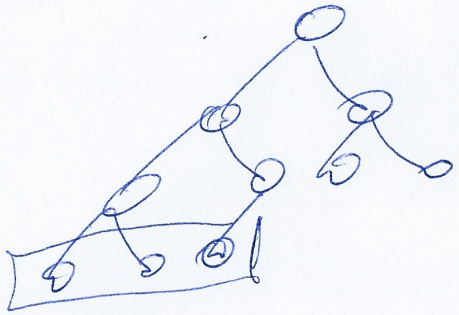


ALL LEVELS ARE COMPLETE BUT THE LOWEST



ELEMENTS $= 2^h - 1$

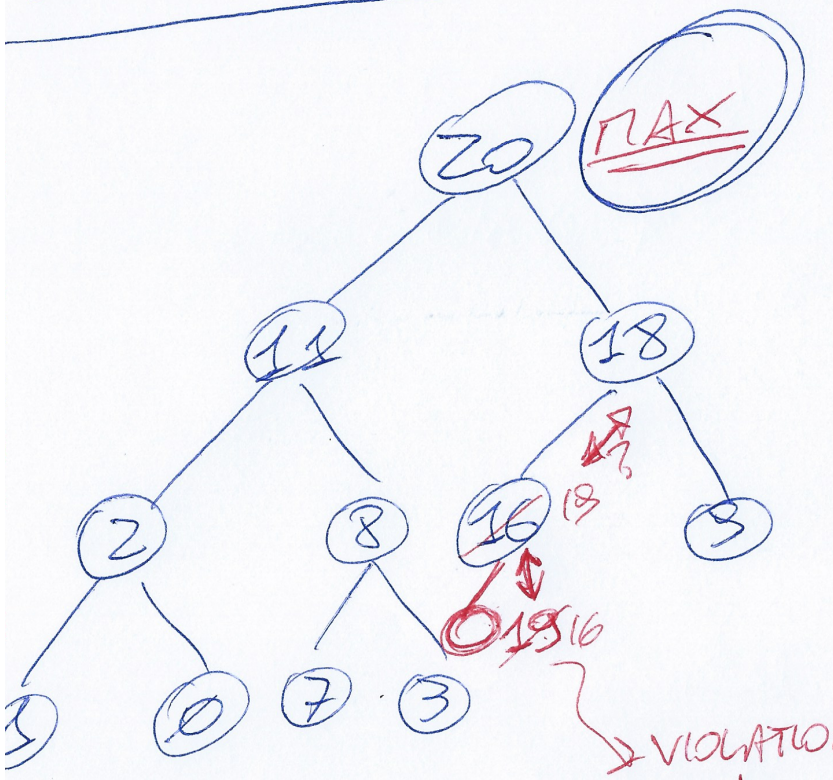
$$\sum_{i=0}^{h-1} 2^i = 2^h - 1$$



- HEAP PROPERTY:

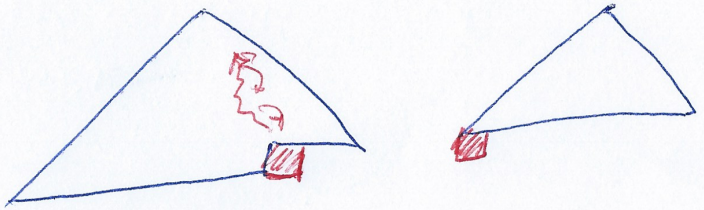
PARENT$(x) \geq x$ ⟹ MAXIMUM IS IN THE ROOT } MAX-HEAP

9 HEAPS

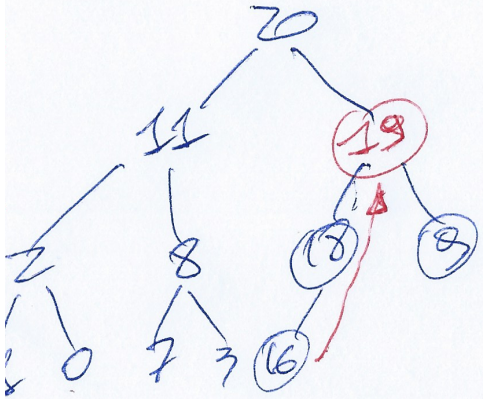


WHERE TO FIND MIN? IN ONE OF THE LEAVES.

- GET MAX $\Theta(1)$ ROOT
- INSERT(x)
  - ADD A LEAF

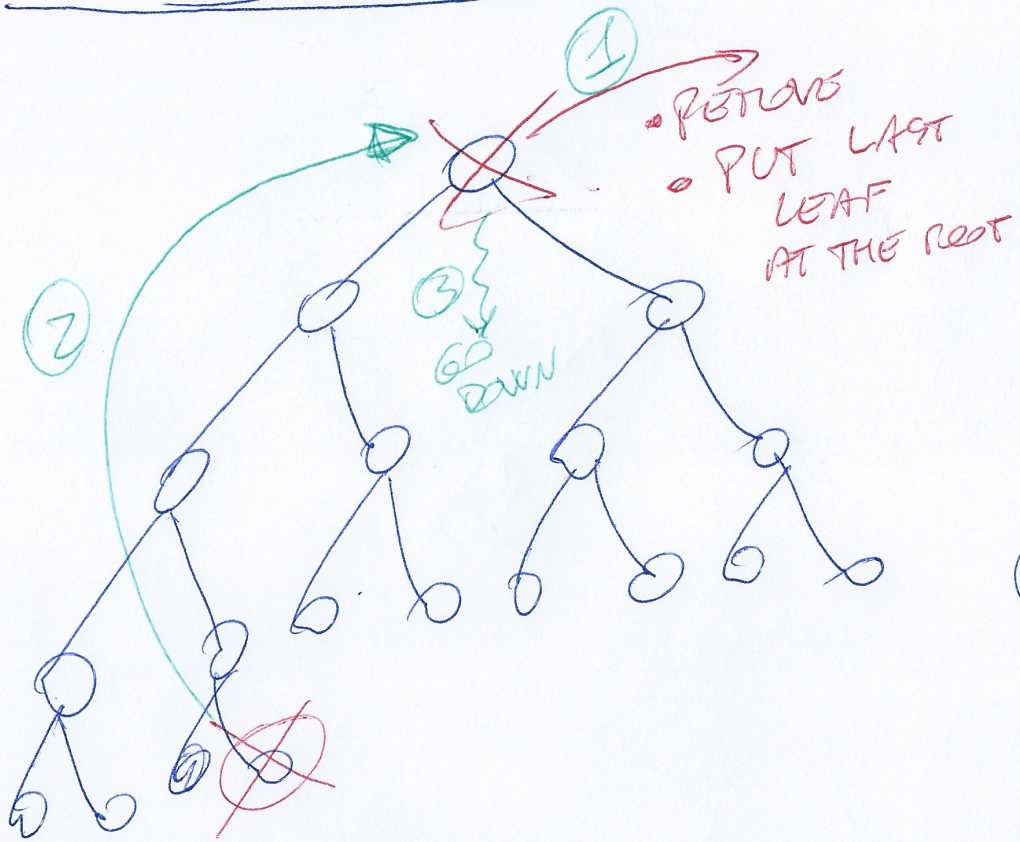


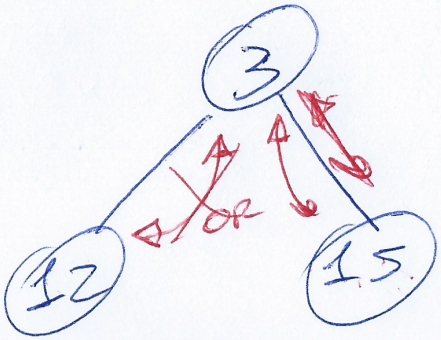
VIOLATION
↓
SWAP PARENT/CHILD
↓
VIOLATION VS PARENT
SWAP
⋮



$\log n$ levels
↓
$\leq \log n$ swaps
$\longrightarrow \Theta(\log n)$

(10) HEAPS. REMOVE MAX

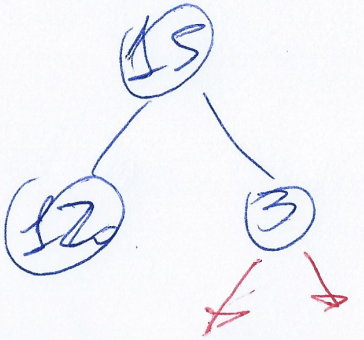


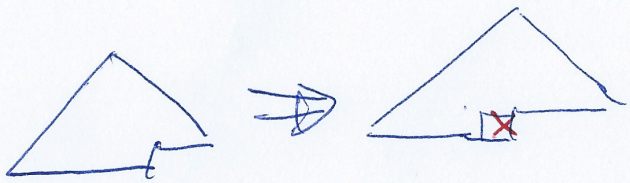


CHOOSE THE LARGEST CHILDREN FOR SWAPPING

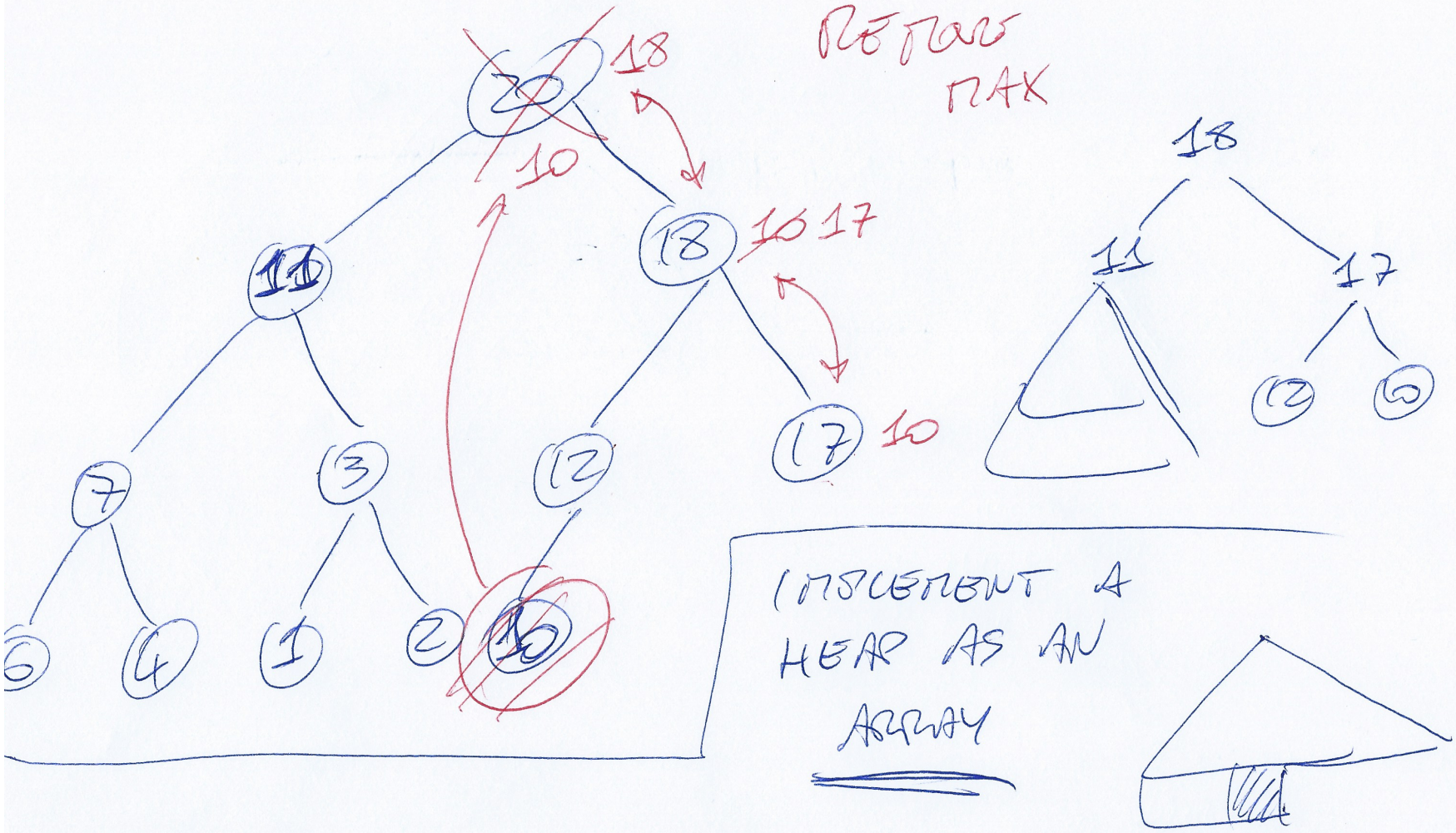
↓

UNTIL IT IS LARGER THAN BOTH CHILDREN





(11) HEAP RETURNS MAX $\Theta(\log n)$



IMPLEMENT A HEAP AS AN ARRAY