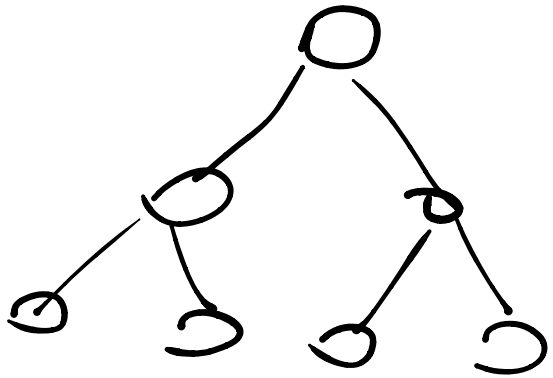
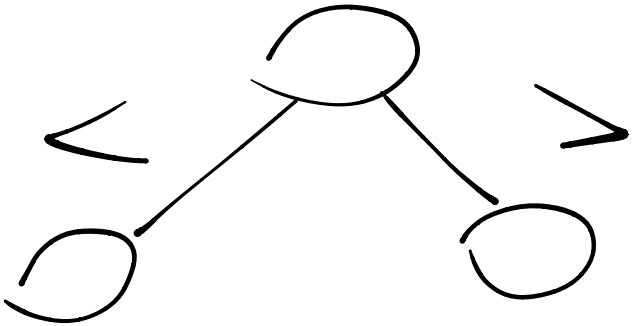
27-10 -> Algoritmi

HEAP -> [ ] (ARRAY)

↕ (HEAPIFY)



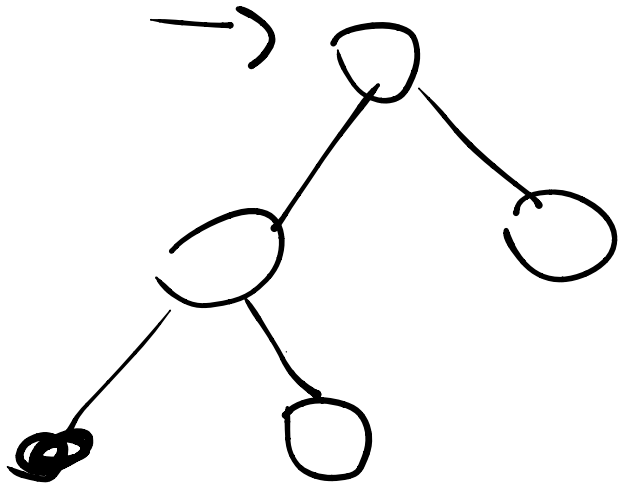
BST ->  PARENT

NODO → ⊗

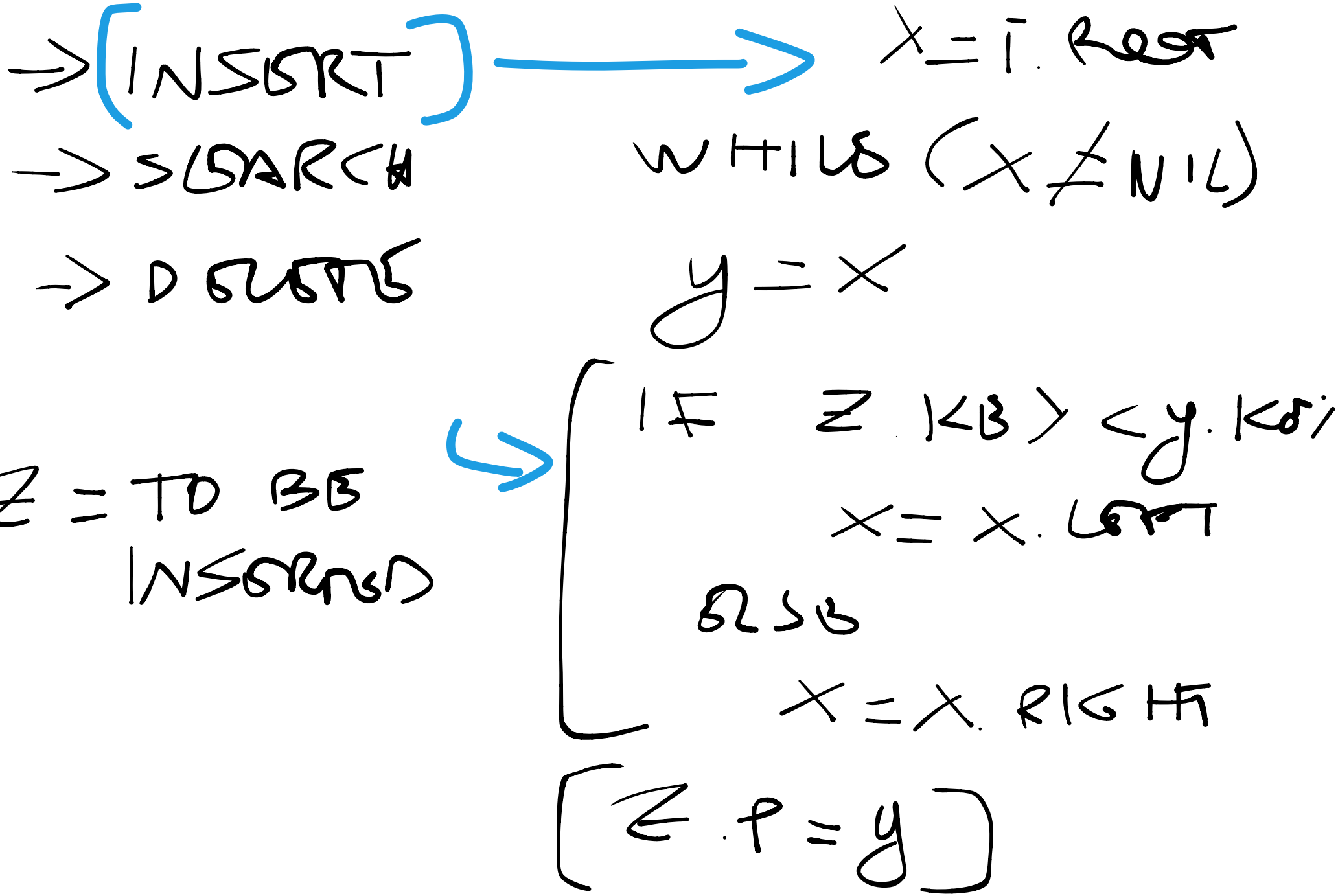
- x.P
- x.KEY
- x.LEFT
- x.RIGHT

ARRICCHIMENTO (CAMPI IN PIÙ)

→ [x.PRED / x.SUCC / x.SIZE]



INFIX (LTR)


INSERT
SEARCH
DELETE
x = T.ROOT
WHILE (x ≠ NIL)
y = x
IF z.KEY < y.KEY
x = x.LEFT
ELSE
x = x.RIGHT
z.p = y
z = TO BE INSERTED

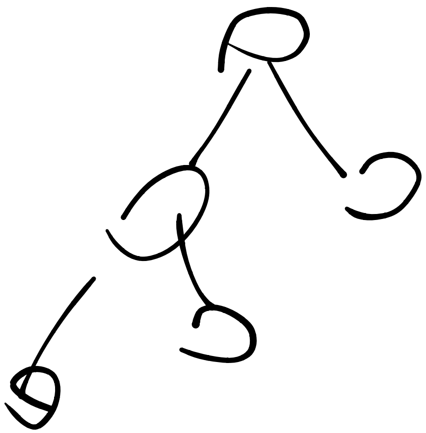
→ $|O(h)|$

IF z.KEY < y.KEY

y.LEFT = z

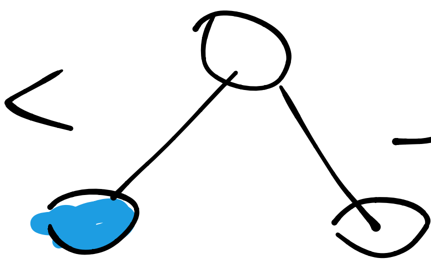
ELSE

y.RIGHT = z



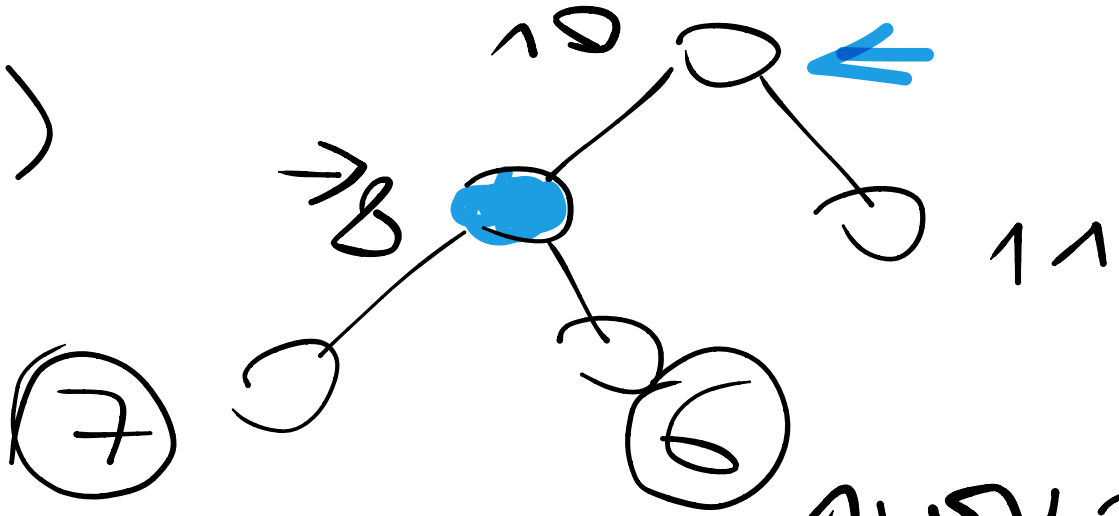
$\rightarrow \Theta(n) = T\left(\frac{n}{2}\right) + C$

(SEARCH) → <



>

(DELETE)

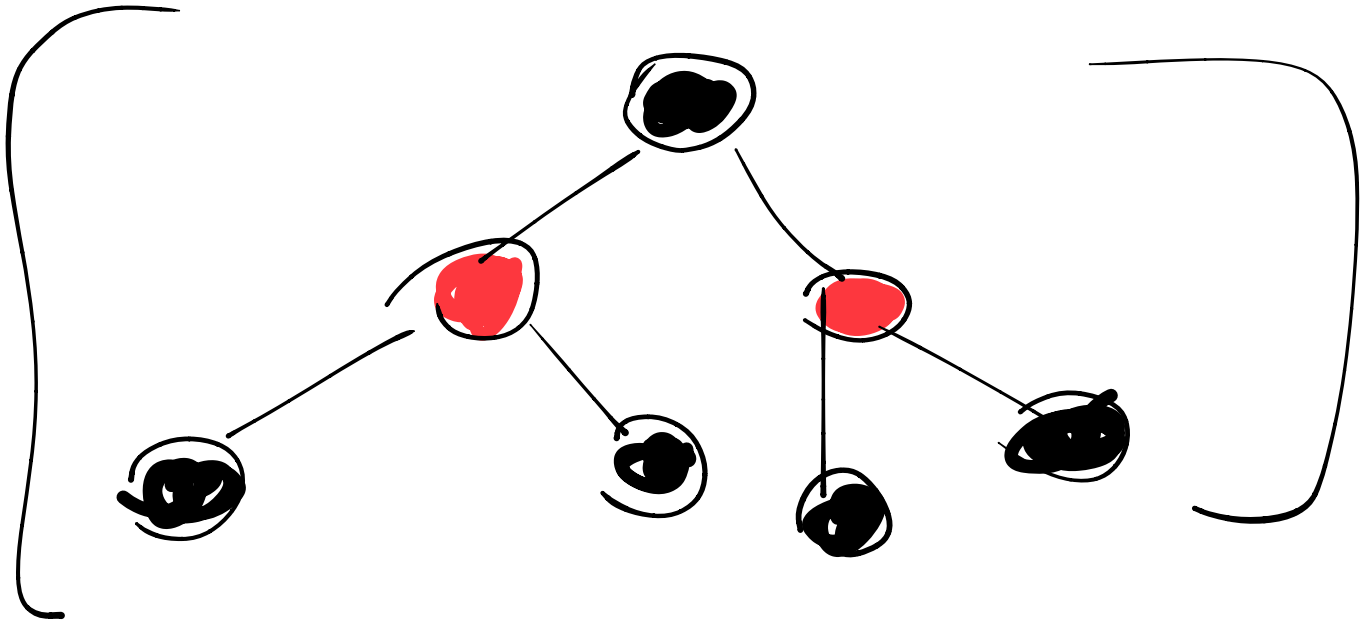


[TRANSPLANT] → AUSI O UN PARENT E DEUX ENFANTS QUI RESTABILISE

RB - TREES

BST

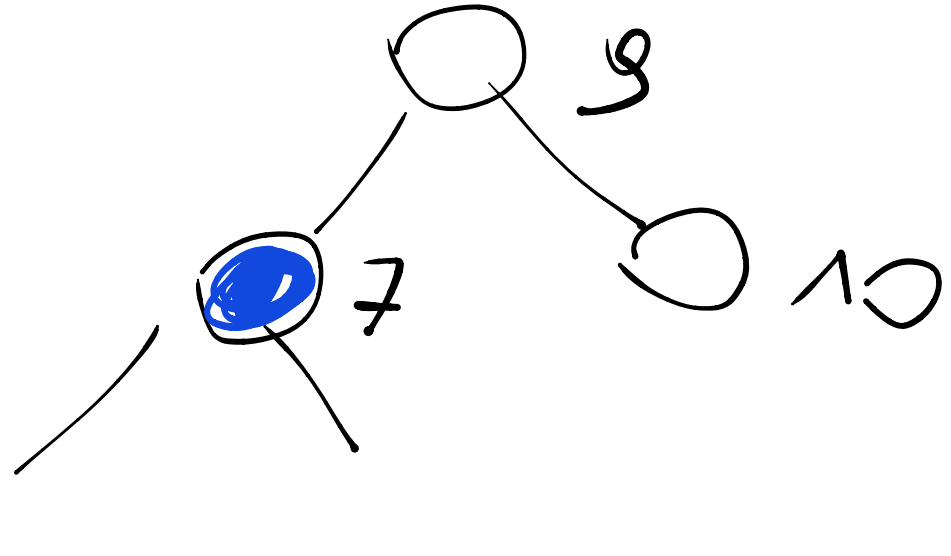
(FOCUS)

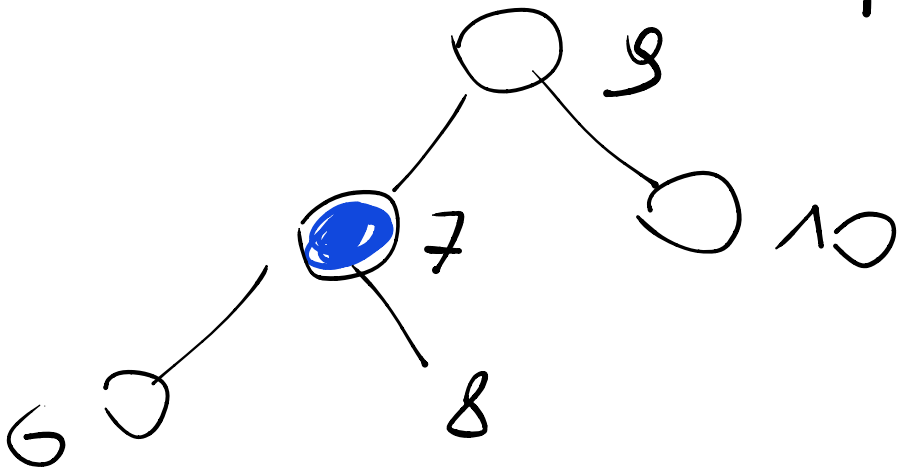


(7.36) → RACCOLTA

PROD(6)

= 7

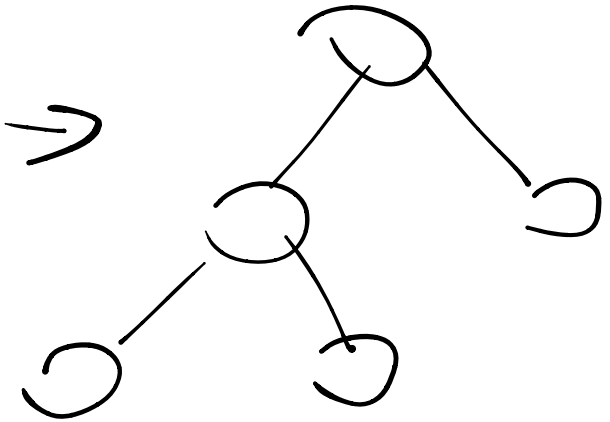





```
PREV(x)
  y = x.ROOT
  WHILE(y)
    IF(x.KEY < y.KEY)
      y = y.LEFT
    IF(x.KEY > y.KEY)
      y = y.RIGHT
    IF(x.KEY = y.KEY)
      RETURN y.P
```

BST -> 7.14

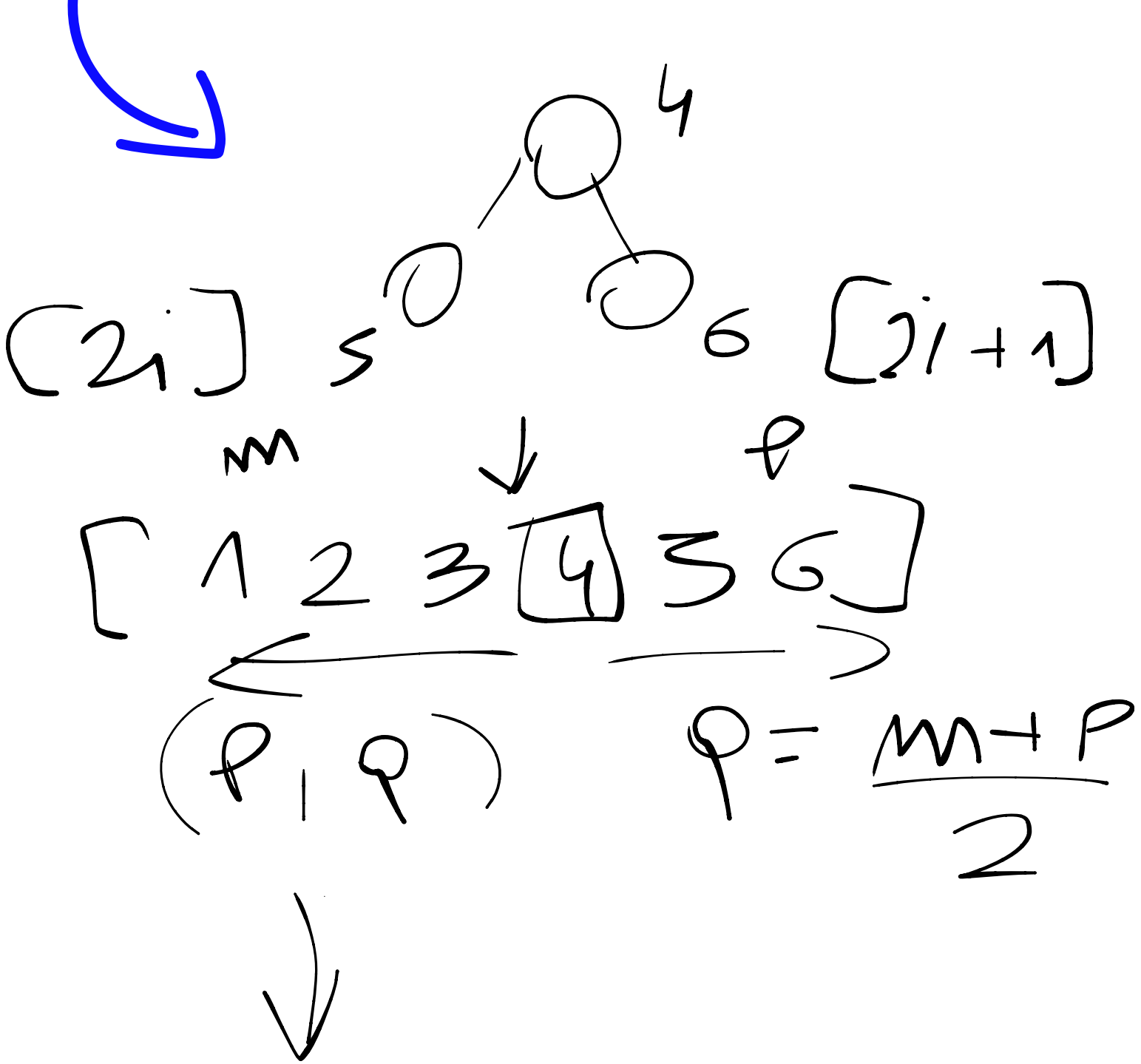
A[1 ... n] =>



[1, 2, 3, 4, 5]

(<) (>)

B. G. [1 2 3 4 5 6]


4
[2i] 5 6 [2i+1]
m
p
[1 2 3 4 5 6]
(p, q)
q = m+p / 2

- BST(A)

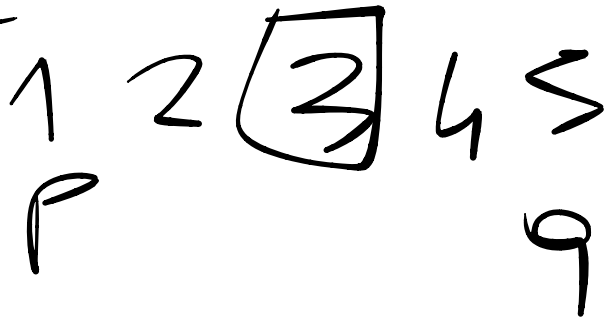
  BST-RSC(A, 1, N)

- BST-RSC(A, P, Q)

  IF (P > Q)

    X = NIL

  M = (P + Q)/2

  X = MKNODE(A[M])

  X.L = BST-RSC(A, P, M-1)

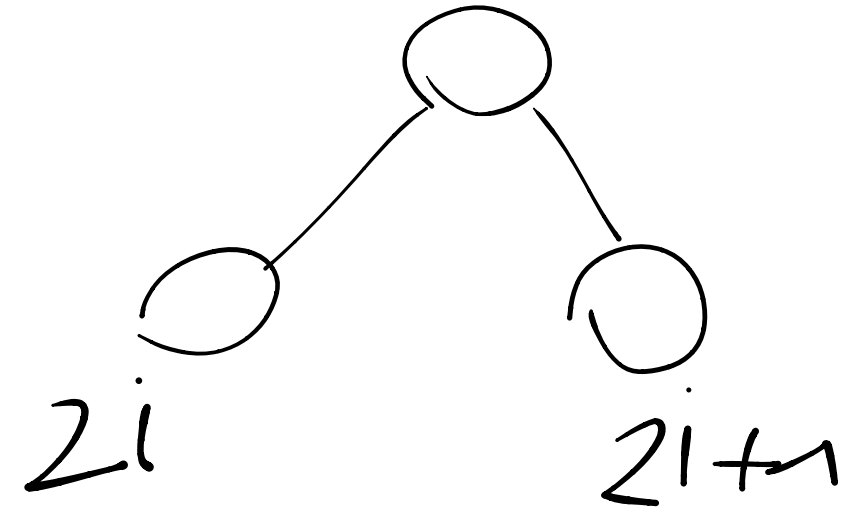
$x.R = BST\text{-}REC(A, m+1, q)$

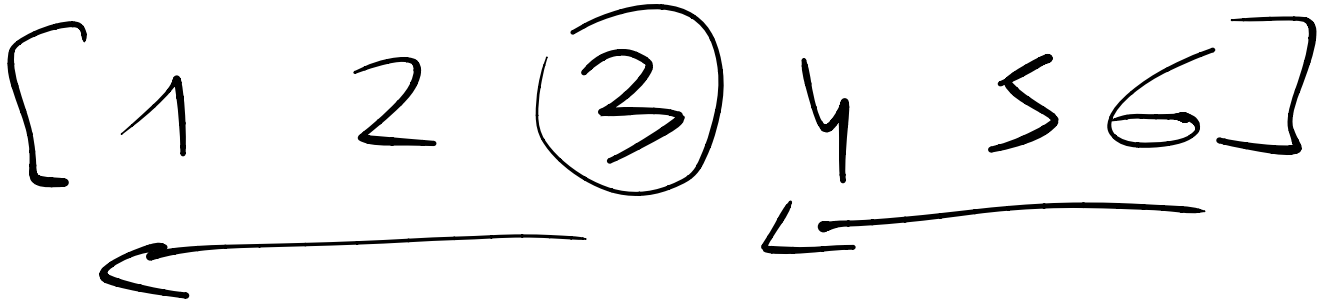
RETURN x

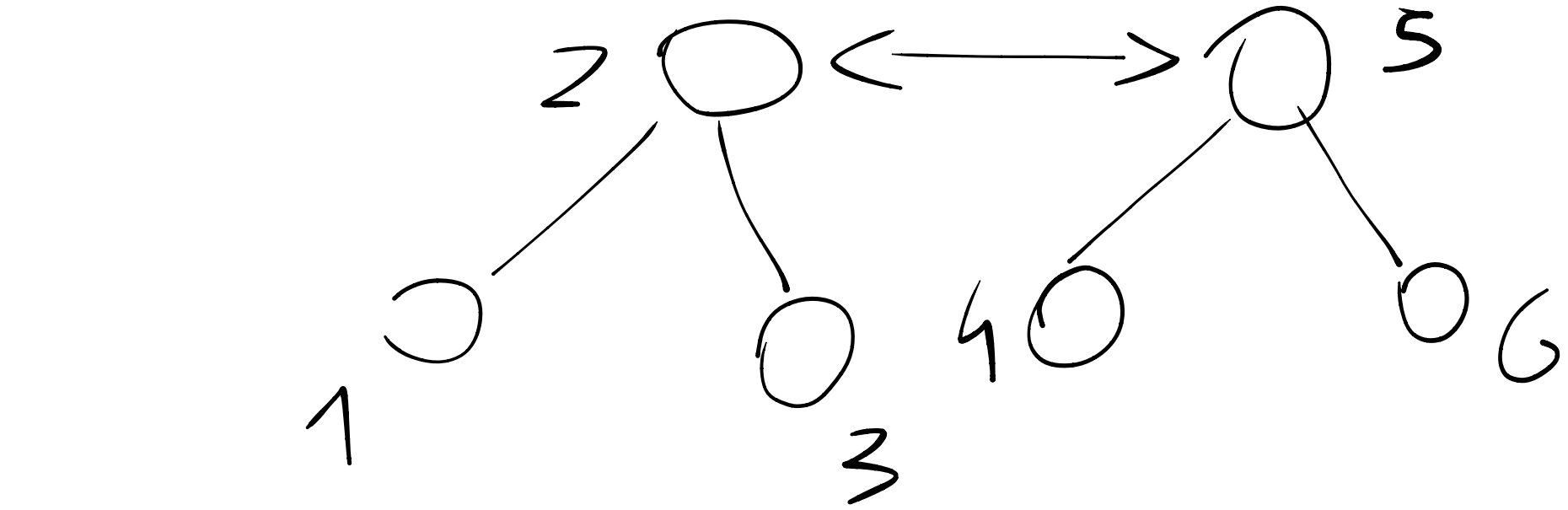
$T(n) = O(n)$

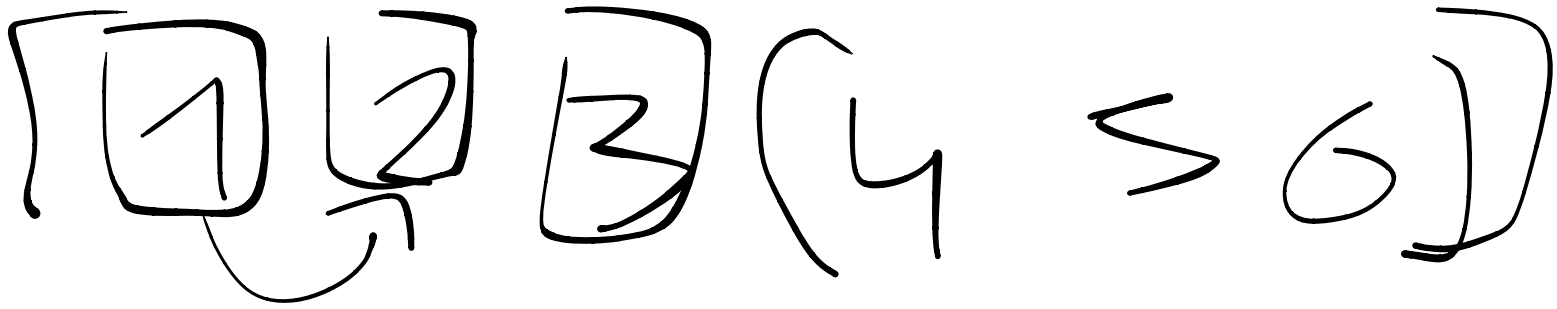
$= 2T\left(\frac{n}{2}\right) + c$

$\Rightarrow BST(A)$

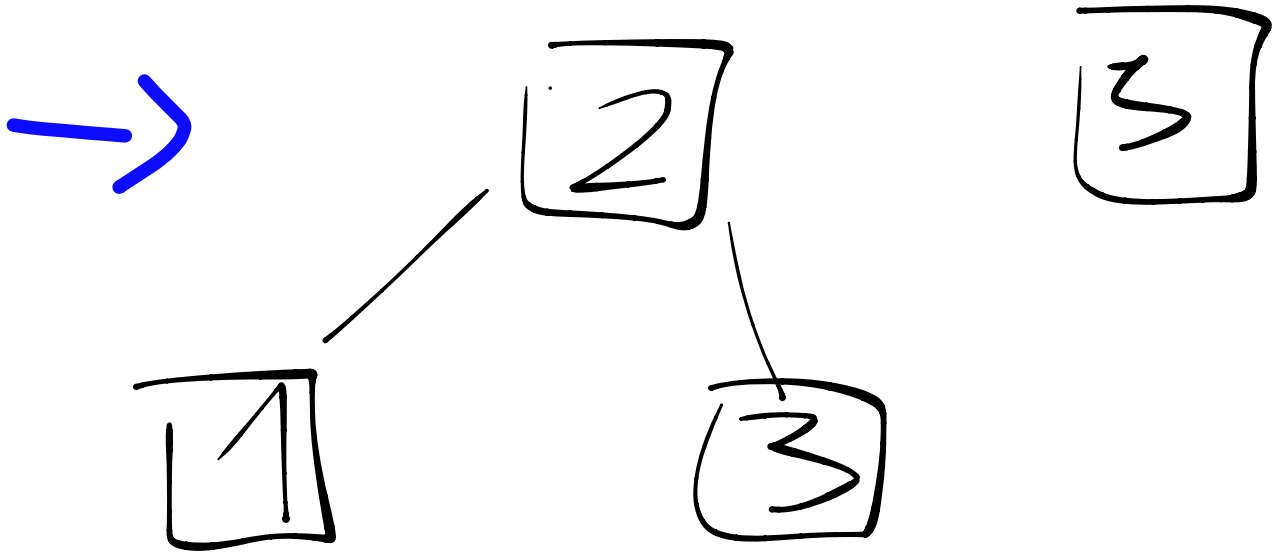




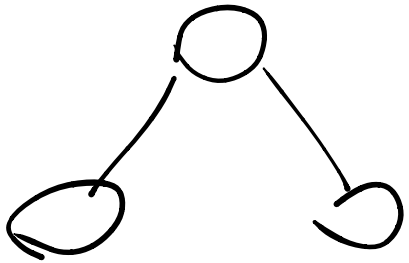




TRANSPLANT

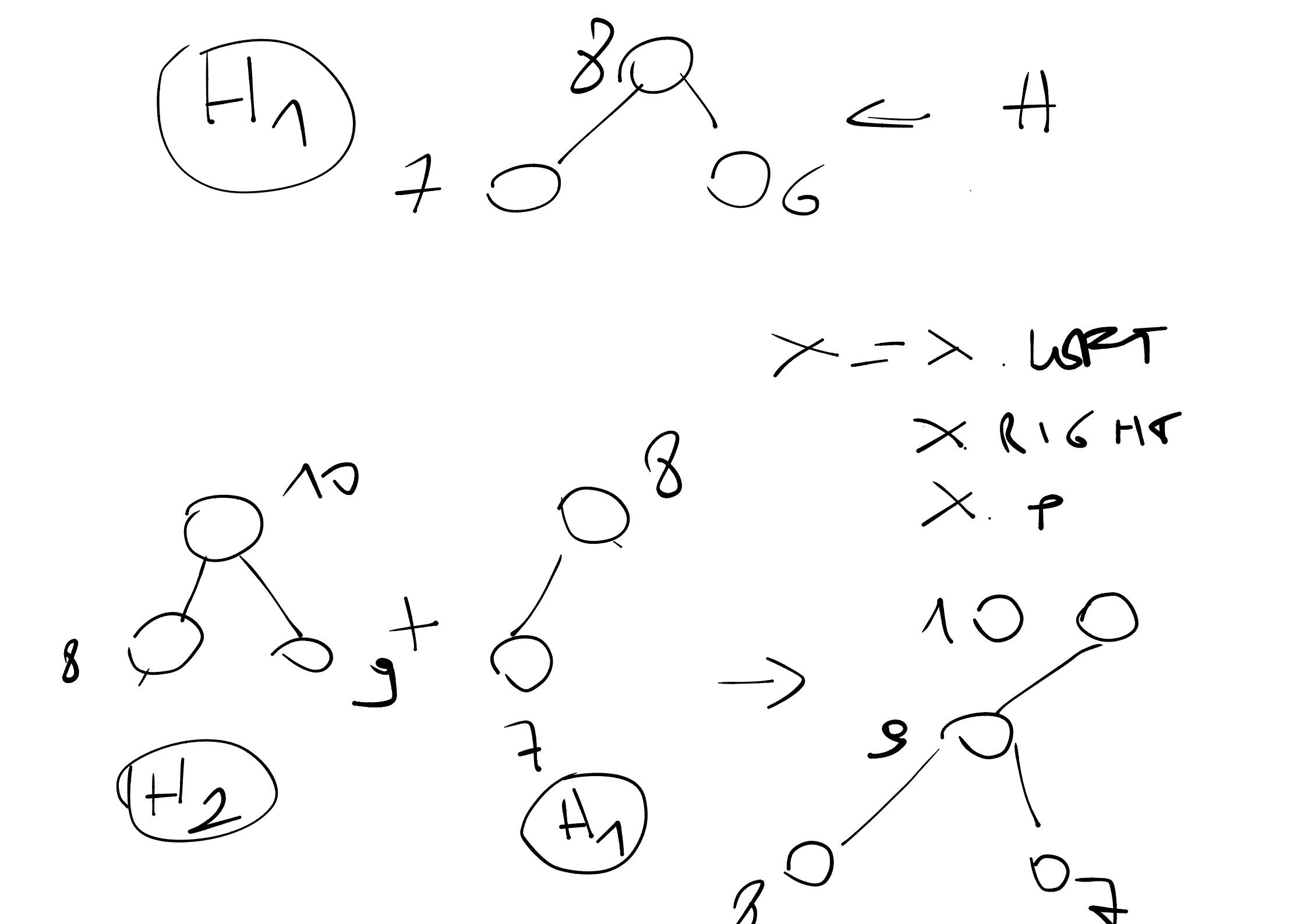


FUSION (H_1, H_2)



MAX-HEAP

$A[1] \geq A[1/2]$


H1
8
7
6
H
x => LEFT
x RIGHT
x. P
10
8
8
9
7
H2
H1
10
9
8
7

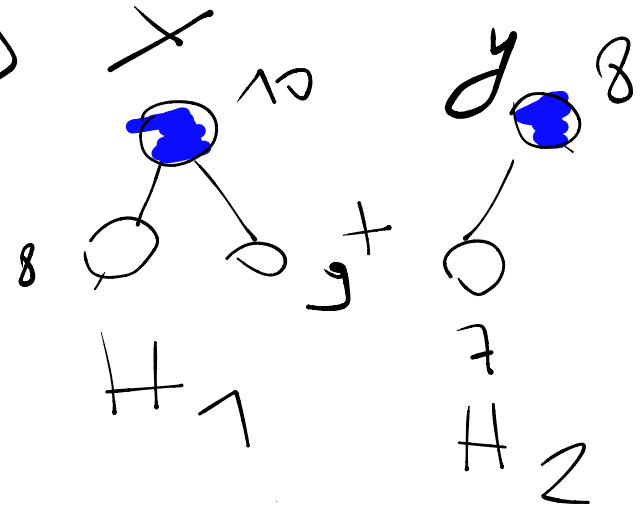
$H_1 + H_2$

FUSION (H_1 , H_2)

-> // SUCCESSOR CASE

H_1 , H_2 -> [x, y] (GIVE)

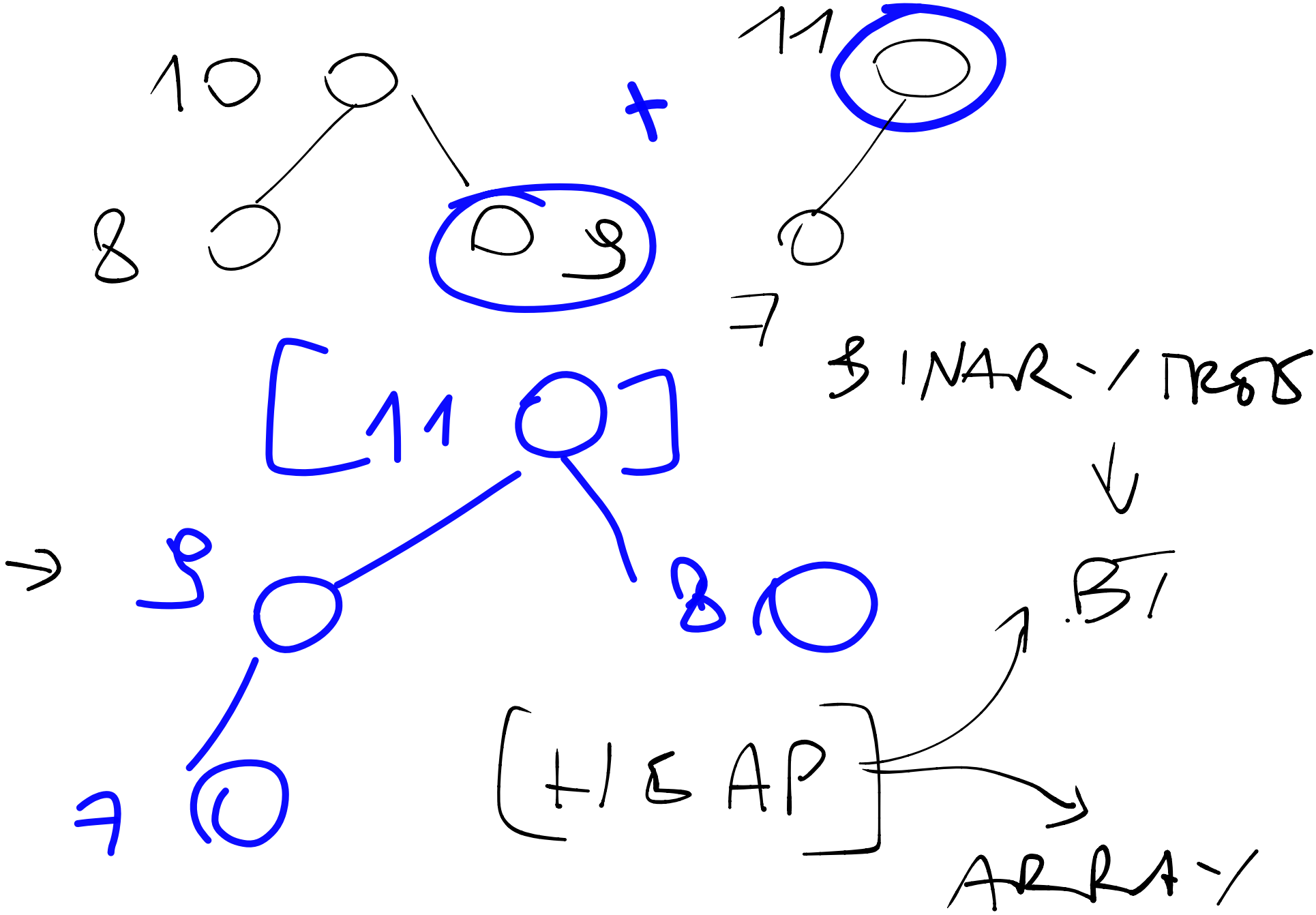
WHILE (H_2 . RIGHT != NIL)



IF (H_2 . LEFT != NIL)

H = MKNODE (H_2 . LEFT)

ELSE


10
8
9
+
11
7
BINARY TREE
[11]
5
8
7
[HEAP]
BT
ARRAY