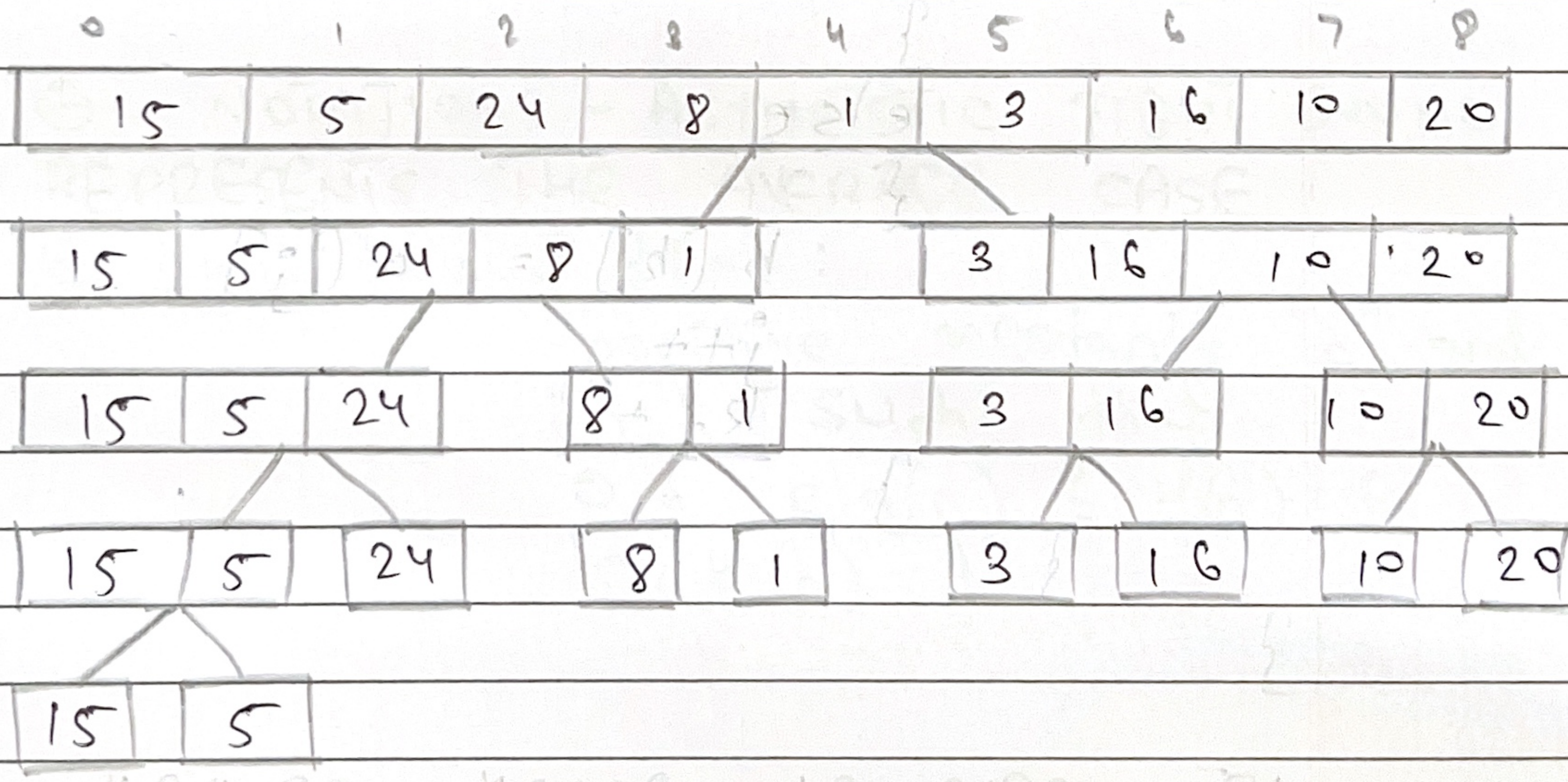
[MERGE SORT]

```
MergeSort (A, lb, up)
{
    if (lb < ub)
    {
        mid = (lb + ub)/2
        MergeSort (A, lb, mid)
        MergeSort (A, mid+1, ub)
        Merge (A, lb, mid, ub)
    }
}
```



```
Merge (A, lb, mid, up)
{
    i = lb
    j = mid + 1
    k = lb

    if (i ≤ mid && j ≤ ub)
    {
        if (a[i] ≤ a[j])
        {
            b[k] = a[i]
            i++
            k++
        }
        else
        {
            b[k] = a[j]
            j++
            k++
        }
    }
}
```

LET ORIGINAL ARRAY BE "A"
LET NEW SORTED ARRAY BE "B"

O - NOTATIONS - ASYMTOTIC UPPER BOUND, REPRESENTS THE WORST CASE

$O(g(n)) = \{ f(n) :$ positive constants "c" and "n_0" such that we have $0 \le f(n) \le cg(n) \}$

Ω - NOTATIONS - ASYMPTOTIC LOWER BOUND, REPRESENTS THE BEST CASE

$\Omega(g(n)) = \{ f(n) :$ positive constants "c" and "no" SUCH that $0 \le cg(n) \le f(n) \}$

Θ - NOTATIONS - ASYMPTOTIC TIGHT BOUND, REPRESENTS THE AVERAGE CASE

$\Theta(g(n)) = \{ f(n) :$ positive constants "c" and "n_0" such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n) \}$

FACTOR	O	Ω	Θ
	ASYMPTOTIC UPPER BOUND	ASYMPTOTIC LOWER BOUND	ASYMPTOTIC TIGHT BOUND
	WORST CASE	BEST CASE	AVG. CASE
	BIG "OH"	BIG "OMEGA"	BIG "THETA"

[RECURRENCE]

[METHODS]

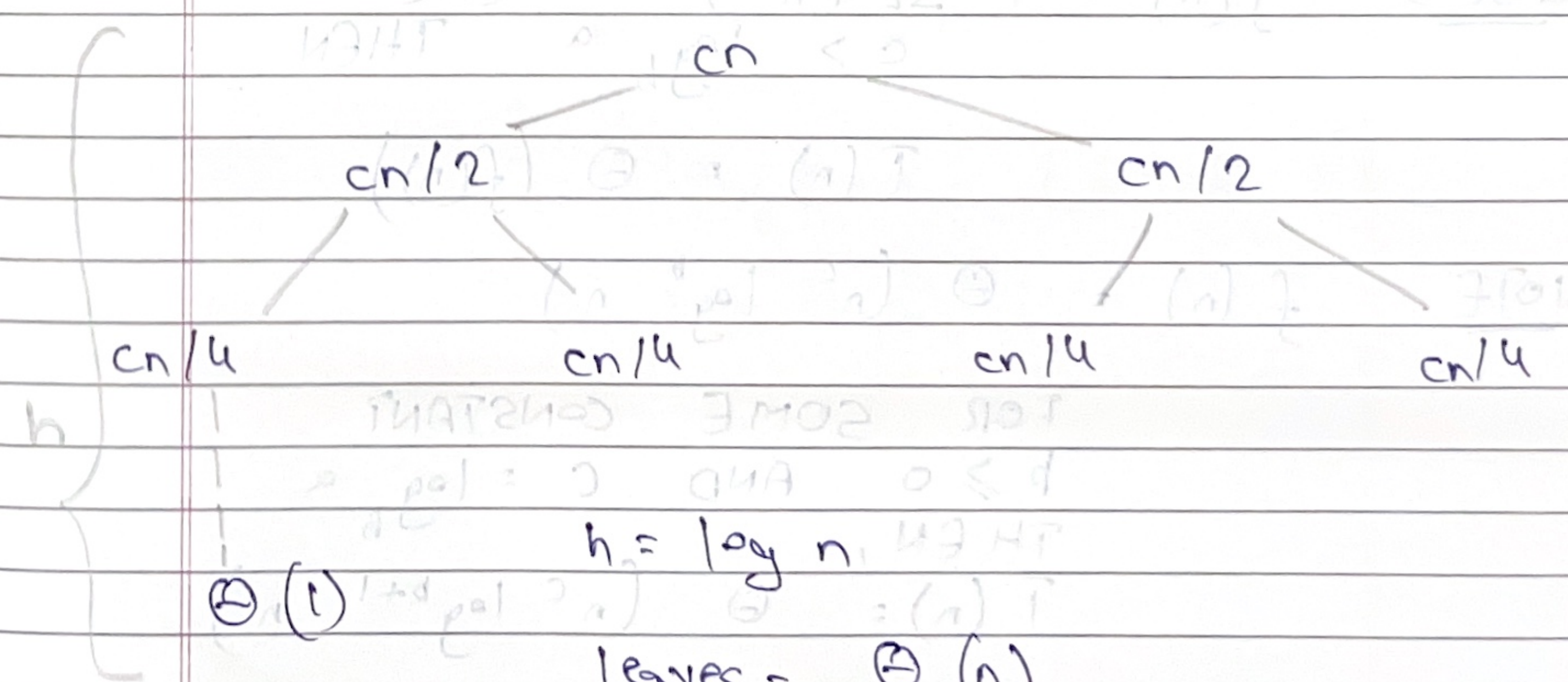
- SUBSTITUTION METHOD
- RECURRENCE TREE
- MASTER THEOREM

[SUBSTITUTION METHOD]

- GUESS THE FORM OF SOLUTION
- VERIFY BY INDUCTION
- SOLVE FOR CONSTANT

Eg.

$$T(n) = 2T(n/2) + cn$$
$$\le 2cn/2 \log(n/2) + n$$
$$= cn \log n - cn \log 2 + n$$
$$= cn \log n - cn + n$$
$$\le cn \log n$$



$h = \log n$

leaves $= \Theta(n)$

TOTAL $= n \log n$

PAGE No.
DATE / /

[MASTER THEOREM]

MASTER METHOD APPLIES TO RECURRENCES OF THE FORM :

$$T(n) = a\,T(n/b) + f(n)$$

WHERE $a \geq 1$, $b > 1$ AND "f" IS ASYMPTOTICALLY POSITIVE

CASE 1 $f(n) = O(n^c)$ WHERE $c < \log_b a$ THEN

$$T(n) = \Theta\left(n^{\log_b a}\right)$$

CASE 2 $f(n) = \Theta(n^c)$ WHERE $c = \log_b a$ THEN

$$T(n) = \Theta(n^c \log n)$$

CASE 3 $f(n) = \Omega(n^c)$ WHERE $c > \log_b a$ THEN

$$T(n) = \Theta(f(n))$$

NOTE $f(n) = \Theta(n^c \log^k n)$

FOR SOME CONSTANT $k \geq 0$ AND $c = \log_b a$

THEN,

$$T(n) = \Theta(n^c \log^{k+1} n)$$

PAGE No.
DATE / /

Eg. $T(n) = 3T(n/2) + n^2$

COMPARING WITH STANDARD FORM,
HERE,

$a = 3$, $b = 2$, $c = 2$

$\log_b a = \log_2 3 = 1.5$

$c = 2$

$c > \log_b a$, ∴ [CASE 3]

$T(n) = \Theta(n^2)$

Eg. $T(n) = T(n/2) + 2^n$

COMPARING WITH STANDARD FORM,
HERE,

$a = 1$, $b = 2$, $c = 1$

$\log_b a = \log_2 1 =$

$c = 1$

$c > \log_b a$ ∴ [CASE 3]

$T(n) = \Theta(2^n)$

Eg. $T(n) = 4T(n/2) + cn$

COMPARING WITH STANDARD FORM,
HERE,

$a = 4,\ b = 2,\ c = 1$

$\log_b a = \log_2 4 =$

$c = 1$

$c \quad \log_b a \ , \therefore$ [CASE]

$T(n) =$

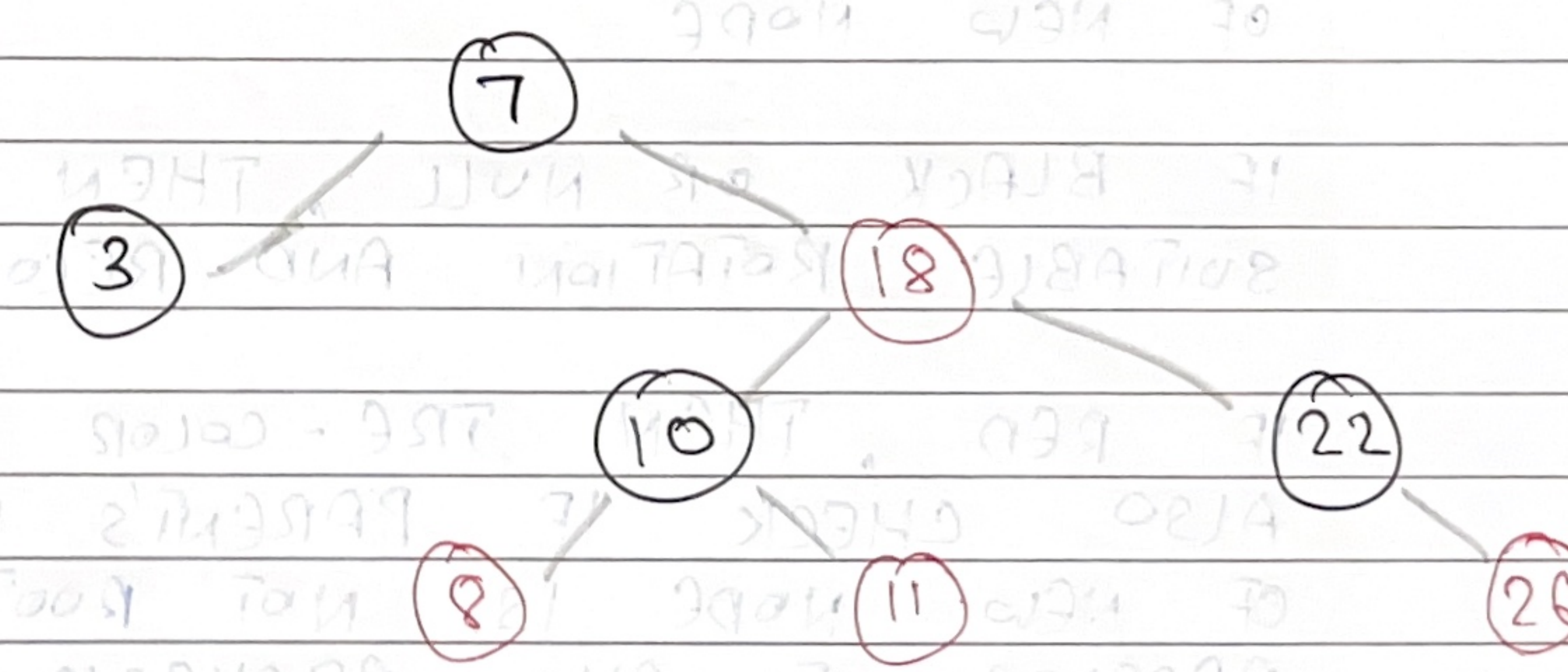
Eg.

[RED - BLACK TREE]

[PROPERTIES]

- EVERY NODE IS EITHER RED OR BLACK
- THE ROOT AND LEAVES ARE BLACK
- IF A NODE IS RED, PARENT IS BLACK
- ALL SIMPLE PATHS FROM ANY NODE "x" TO A DESCENDANT LEAF HAVE THE SAME NUMBER OF BLACK NODES THAT IS "BLACK-HEIGHT (x)"

Eg.



PAGE No. DATE / /

Eg. 10, 18, 7, 15, 16, 30, 25, 40, 60, 2, 1, 70

IF THE TREE IS EMPTY, CREATE THE ROOT NODE AS BLACK
ELSE CREATE NEW NODE AS LEAF NODE AS RED

IF PARENT OF NEW NODE IS BLACK THEN EXIT

IF PARENT OF NEW NODE IS RED CHECK COLOR OF PARENT'S SIBLING OF NEW NODE

IF BLACK OR NULL, THEN PERFORM SUITABLE ROTATION AND RECOLOR.

IF RED, THEN RE-COLOR AND ALSO CHECK IF PARENT'S PARENT OF NEW NODE IS NOT ROOT THEN RECOLOR IT AND RECHECK

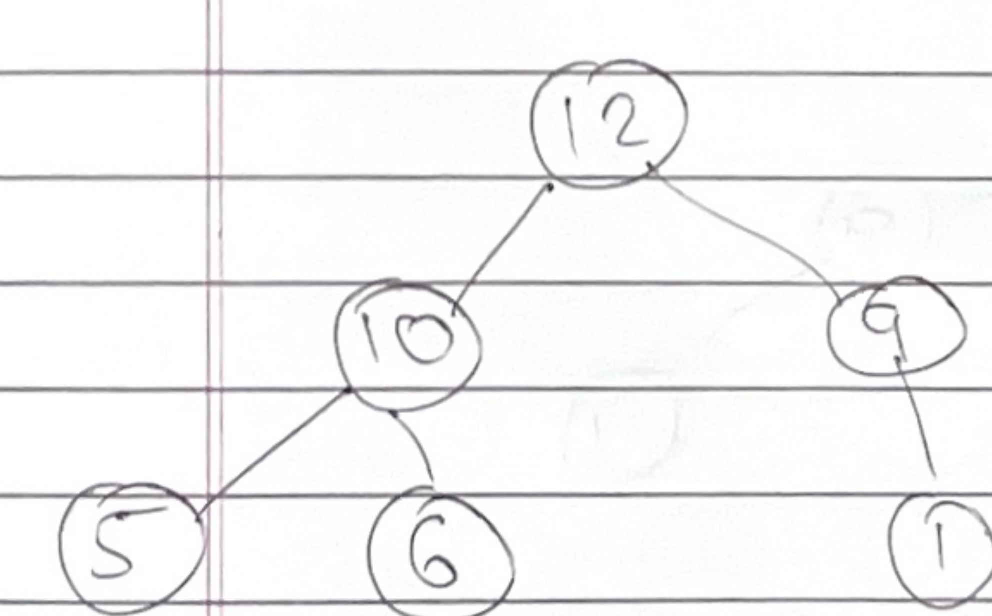
PAGE No. DATE / /

[HEAP SORT]

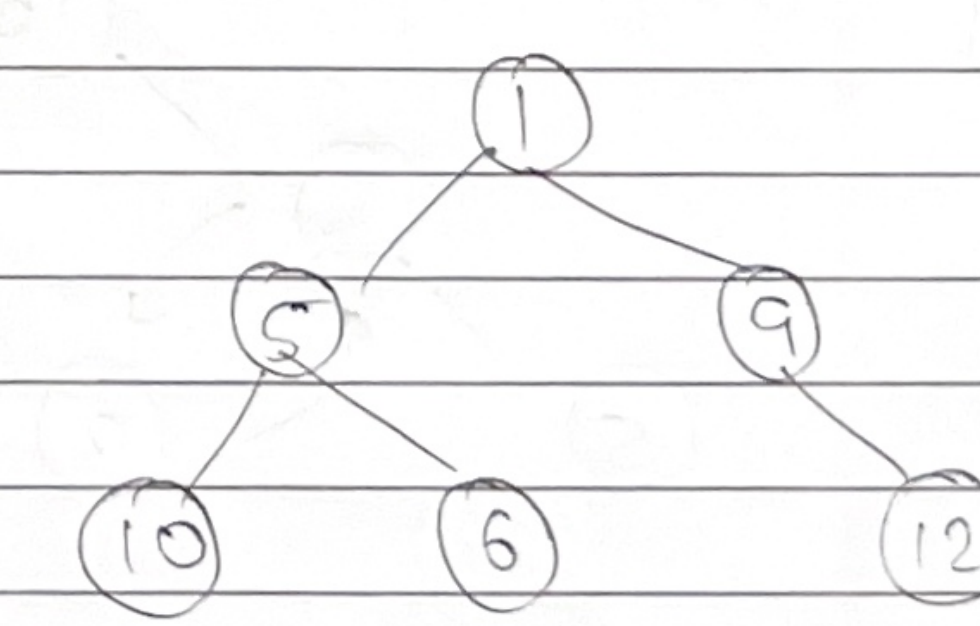
- A SORTING TECHNIQUE THAT USES A COMPLETE BINARY TREE IN WHICH ALL LEVELS ARE COMPLETELY FILLED EXCEPT LAST LEVEL

IF INDEX OF ANY ELEMENT IS "i"
- THE ELEMENT IN INDEX "2i+1" WILL BECOME LEFT CHILD
- INDEX "2i+2" WILL GIVE THE RIGHT CHILD
- ALSO PARENT OF ANY ELEMENT

[MAX HEAP]

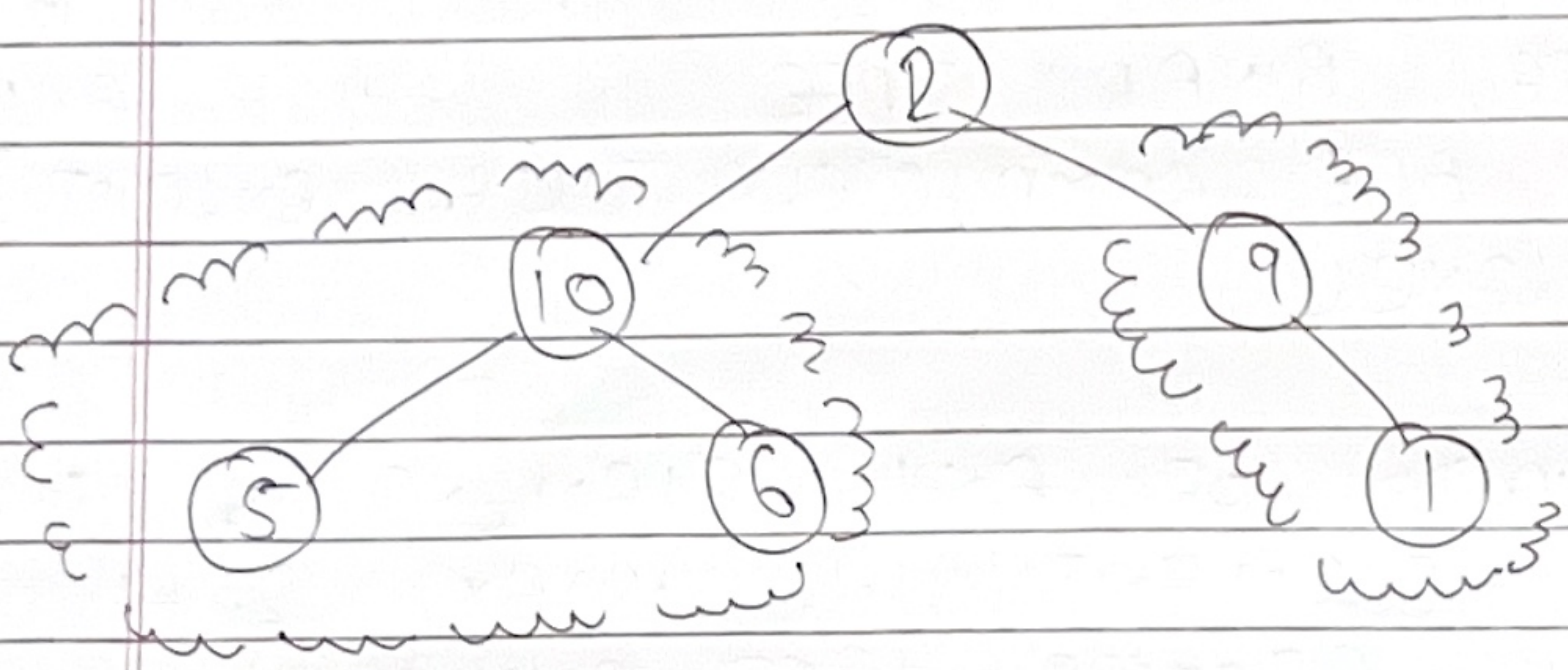


[MIN HEAP]

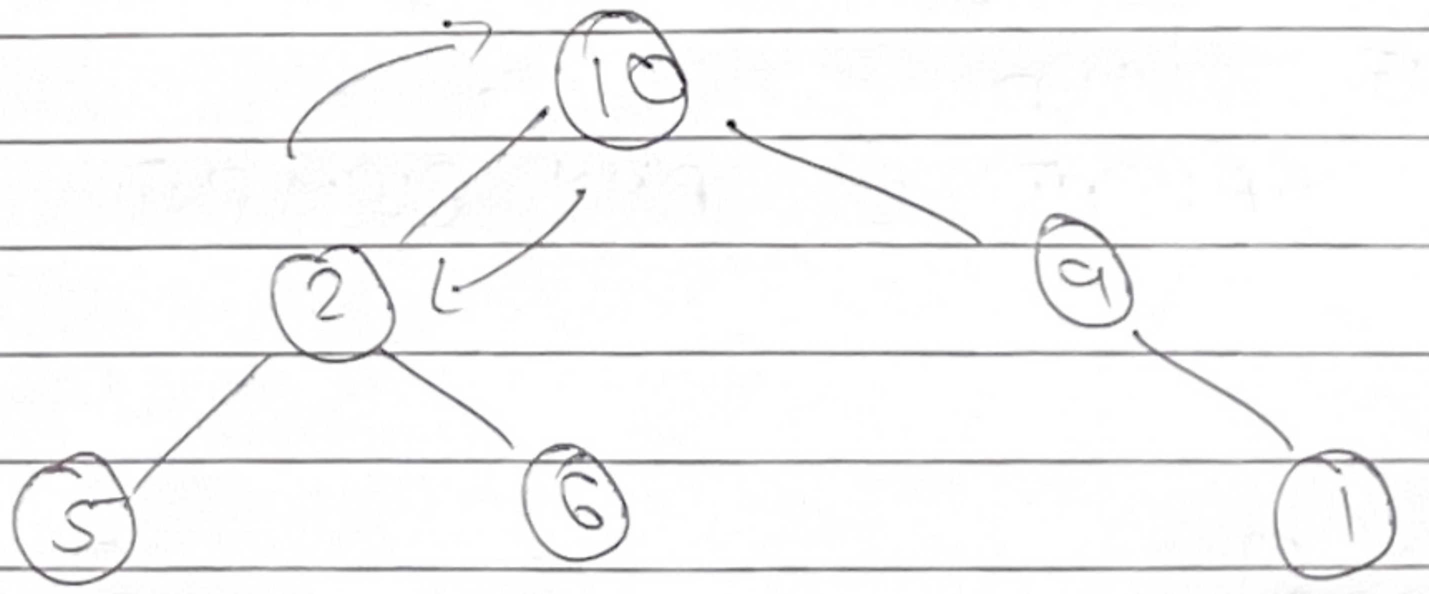


PAGE No. DATE / /

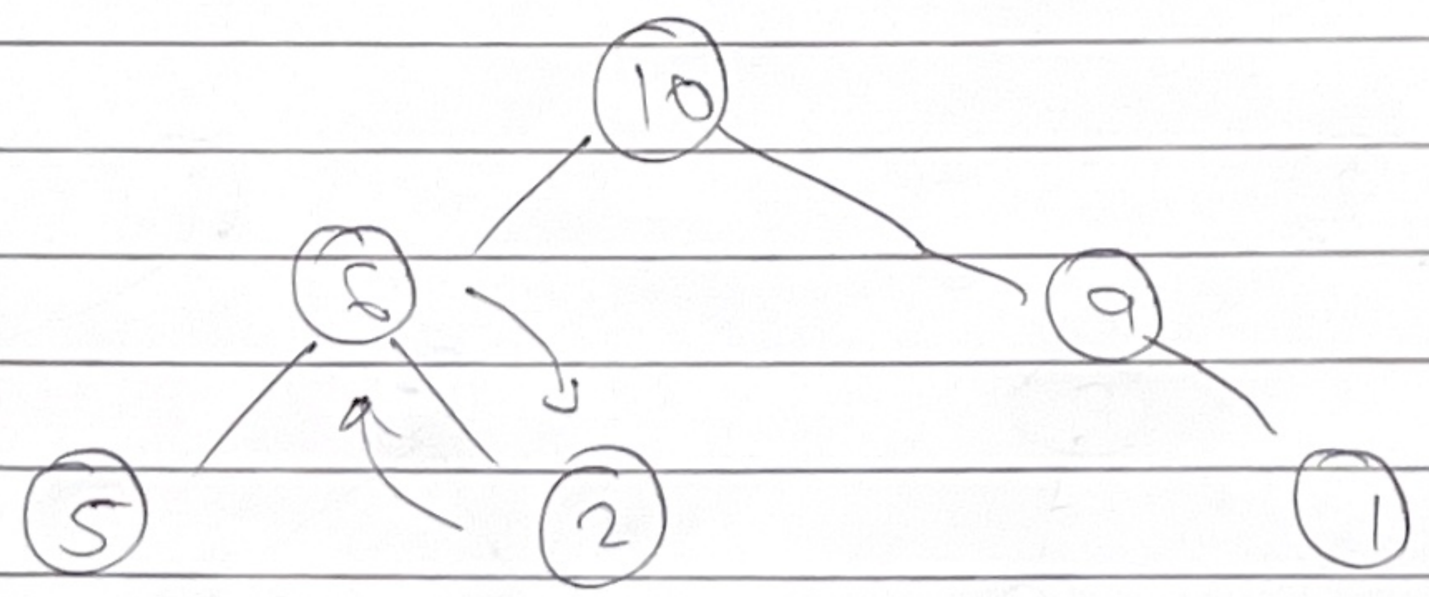
HOW TO HEAPIFY
IF SUBS ARE MAX



1



2

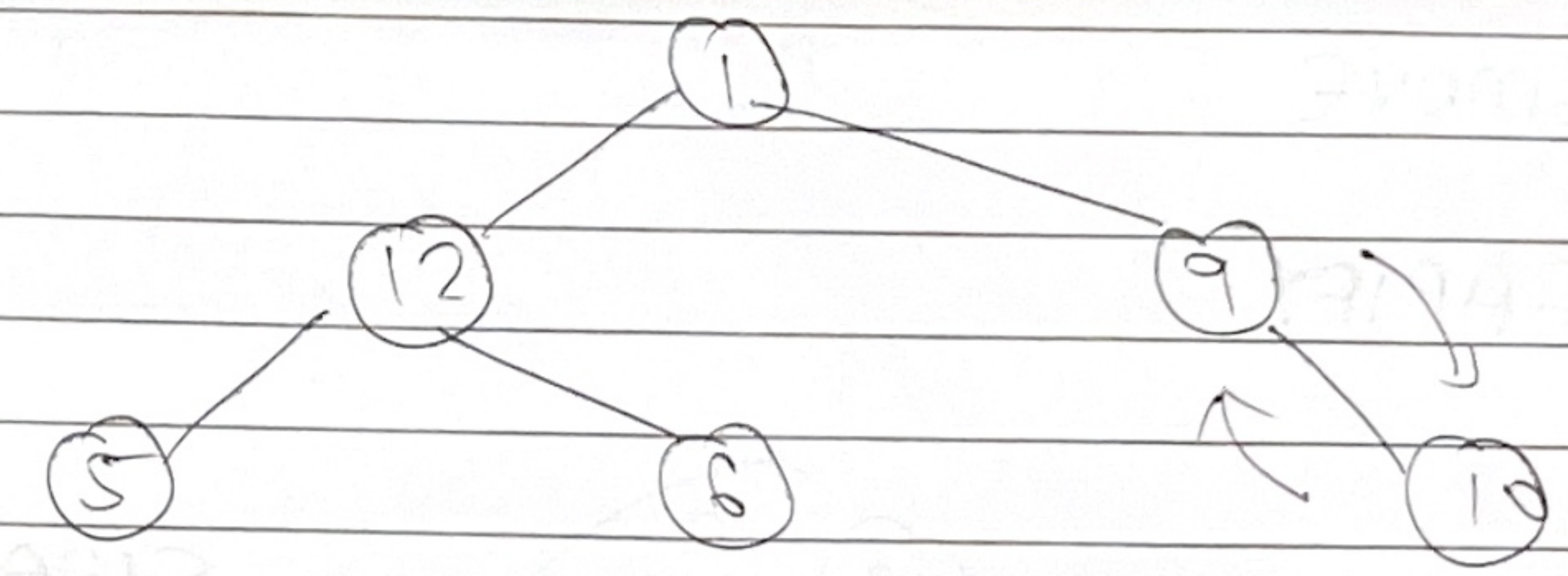


PAGE No. DATE / /

0	1	2	3	4	5
1	12	9	5	6	10

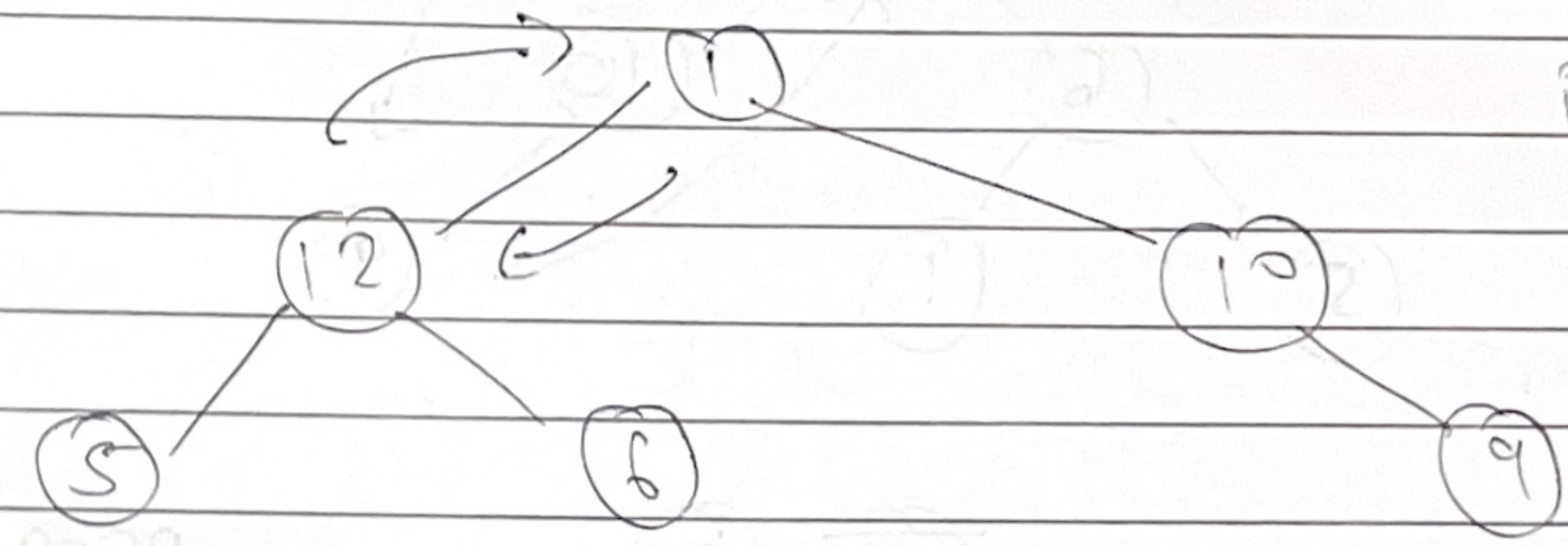
n = 6

i = 6/2 - 1 = 3 - 1 = 2



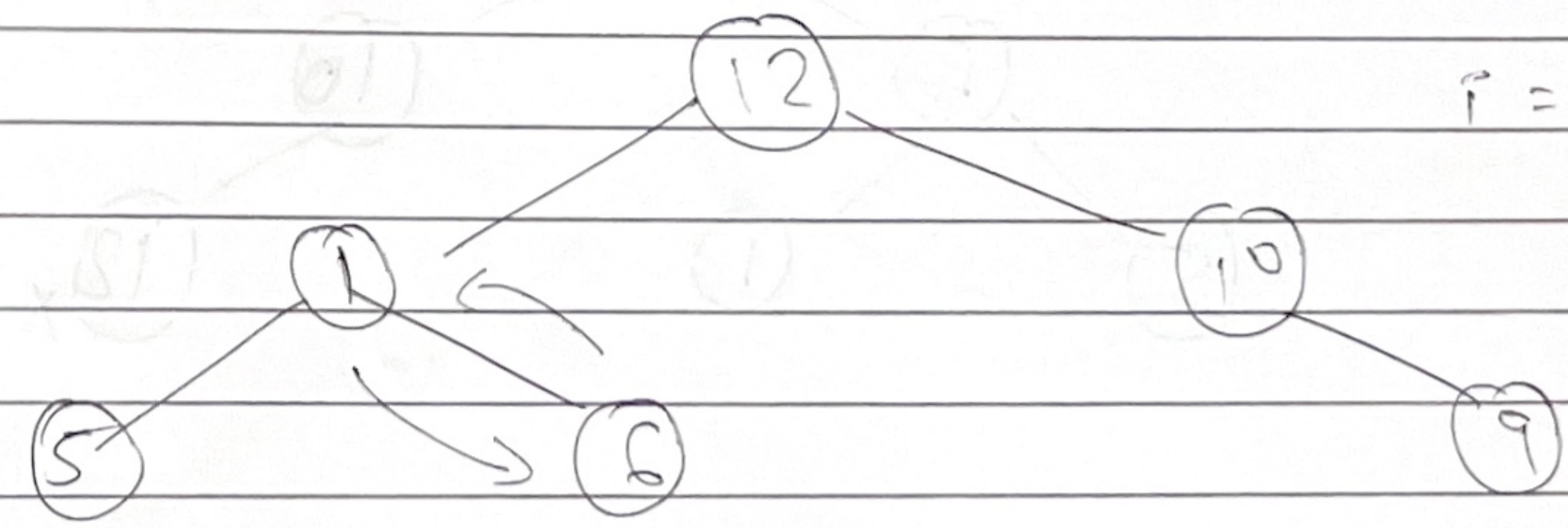
1

i = 2



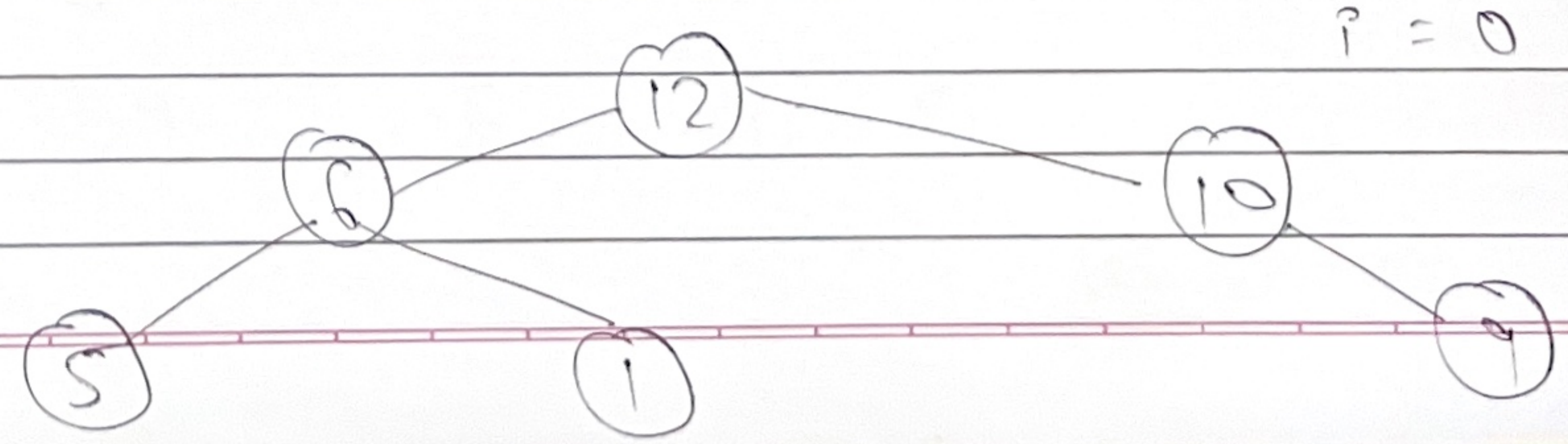
2

i = 1



3

i = 0



[WORKING]

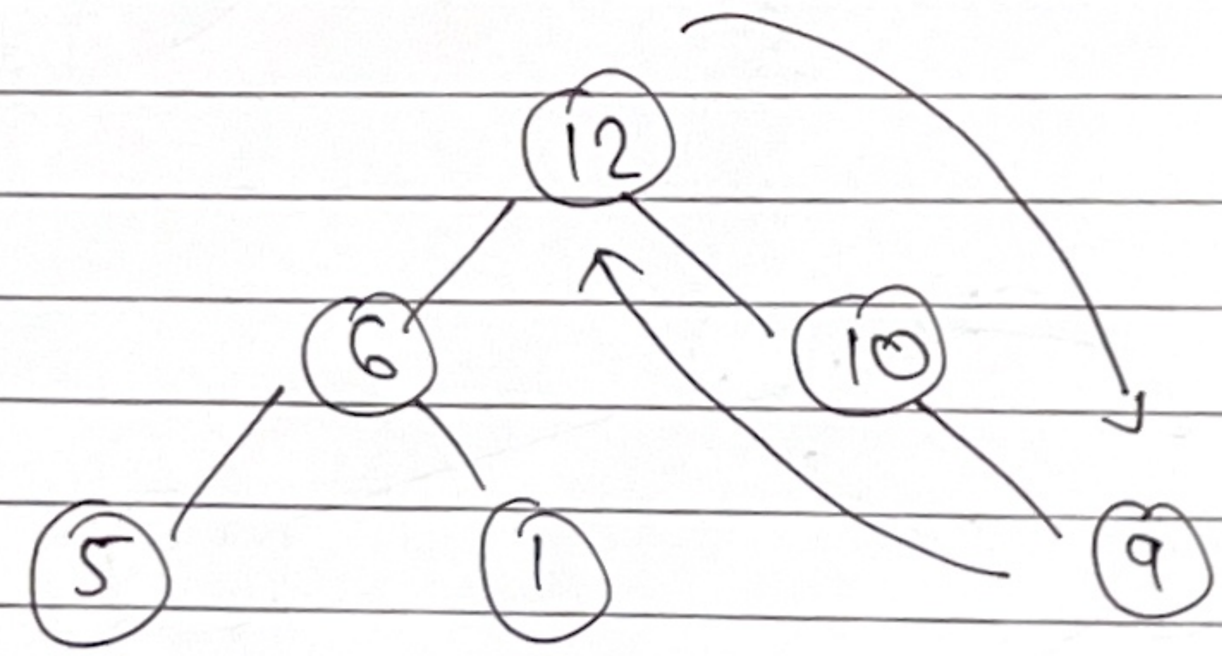
MAKE SURE THE TREE IS IN THE MAX-HEAP PROPERTY

SWAP

REMOVE

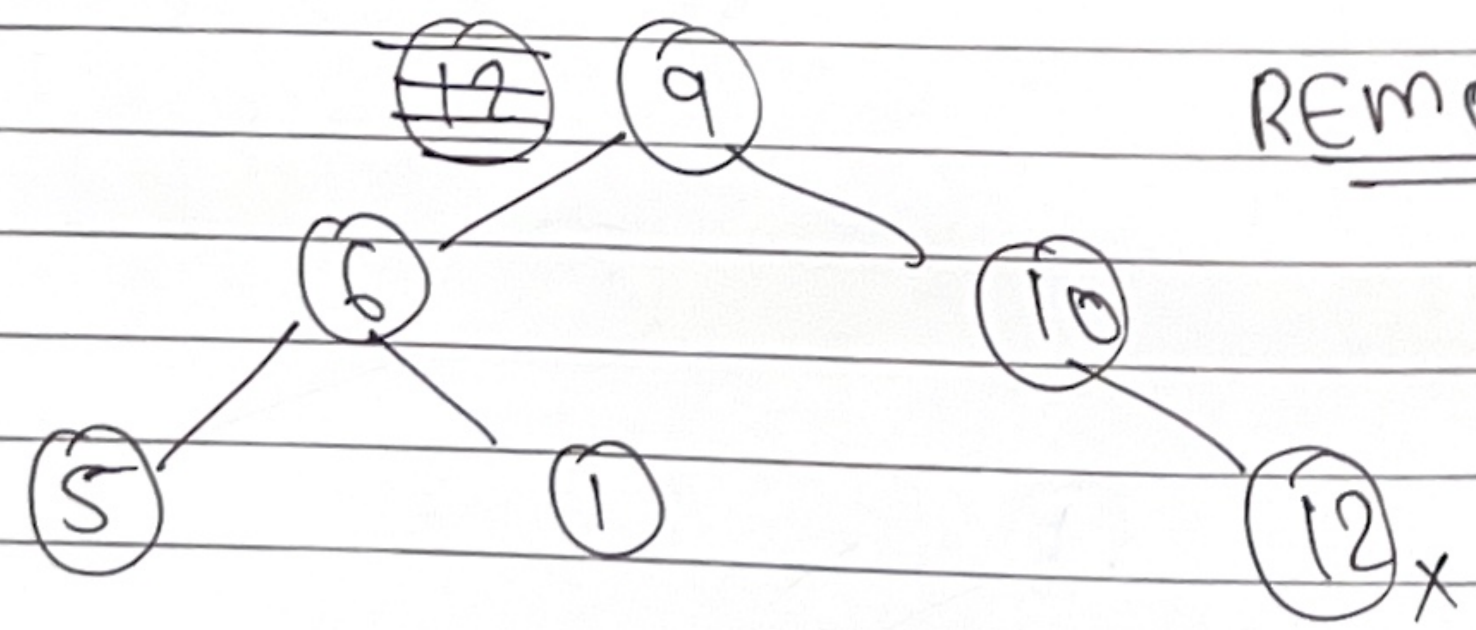
HEAPIFY

1]



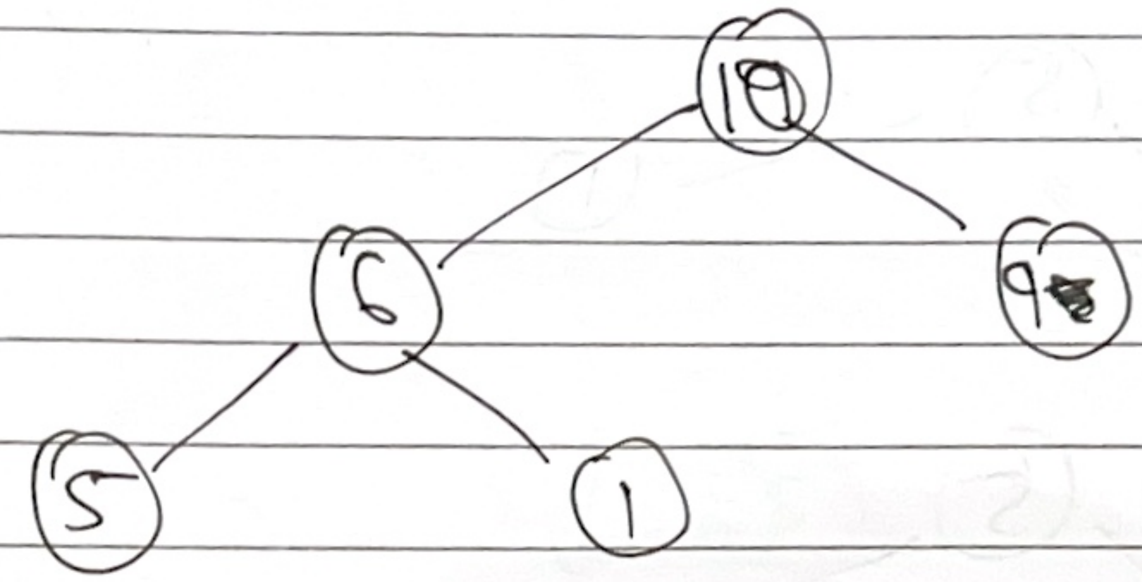
SWAP

2]



REMOVE

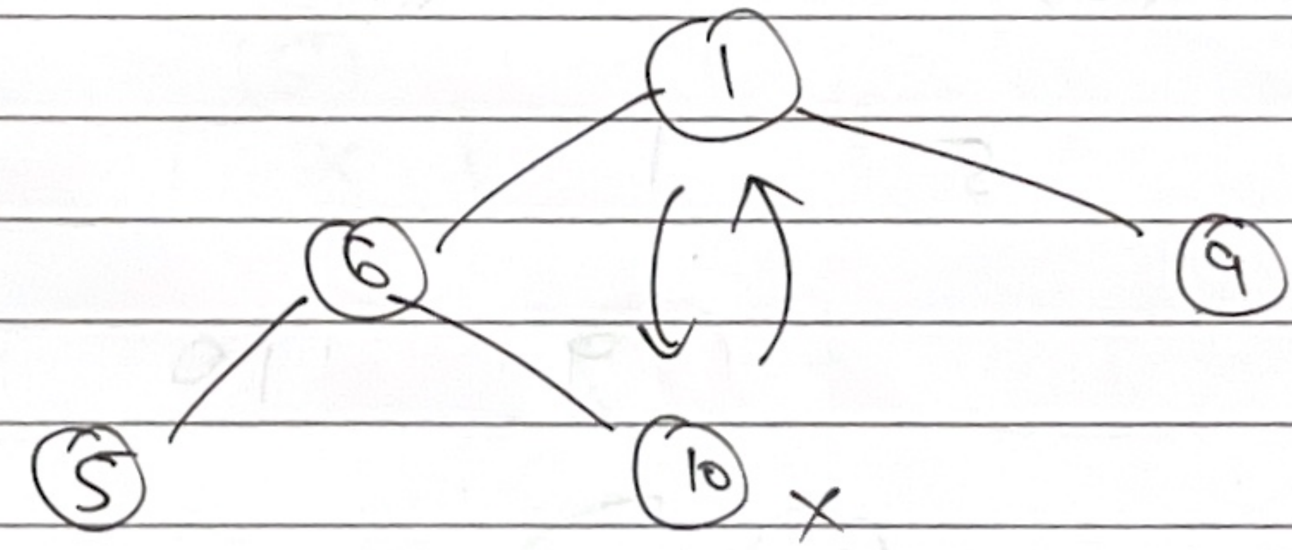
3]



HEAPIFY

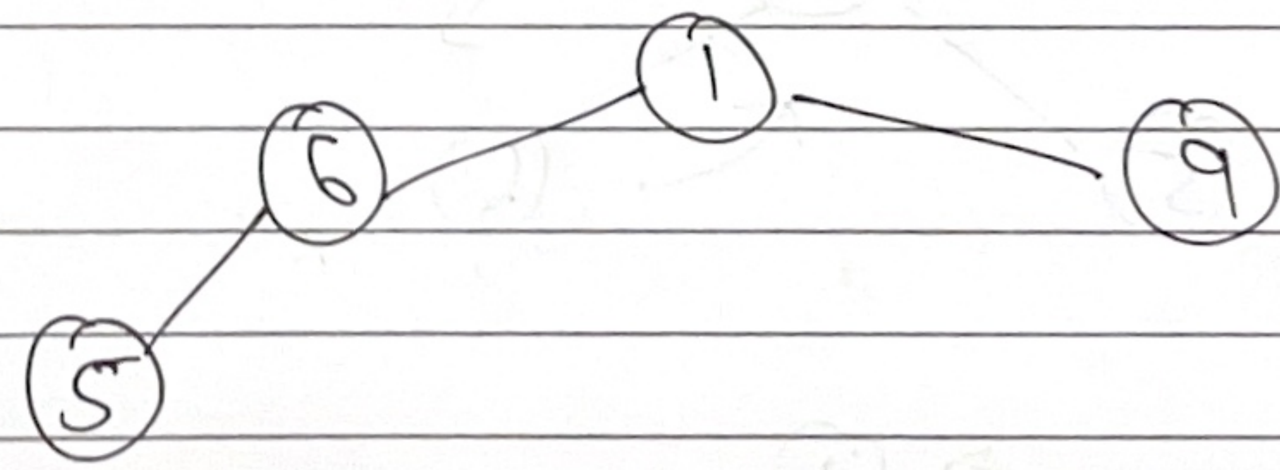
10	6	9	5	1	
					12

1]



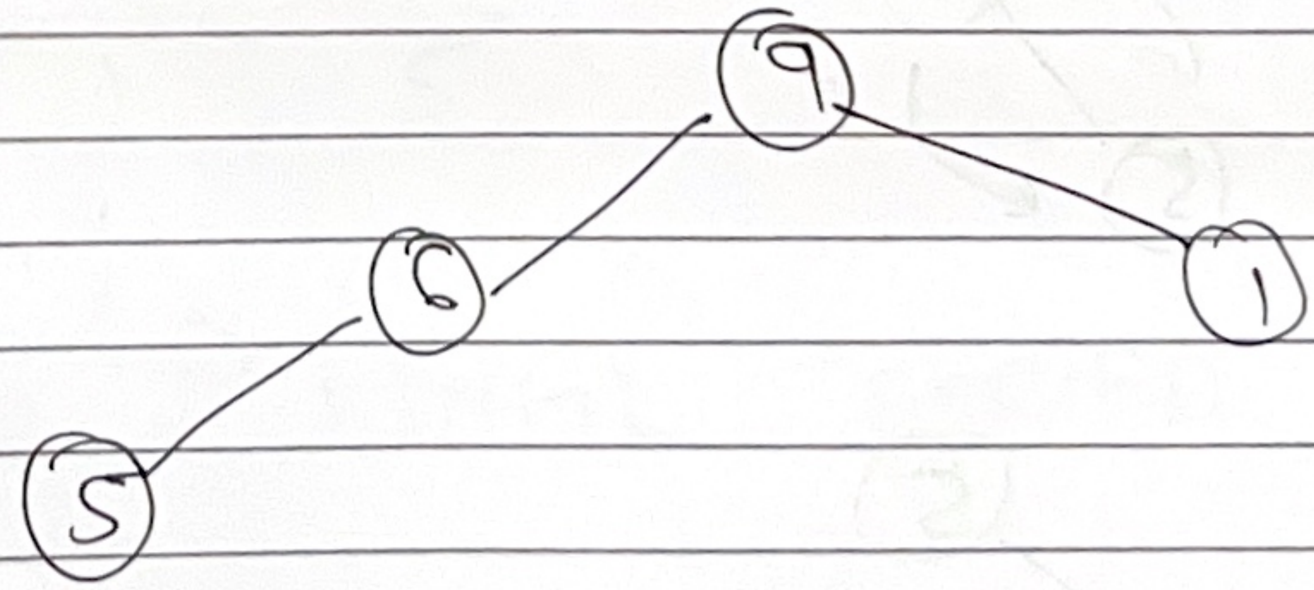
SWAP

2]



REMOVE

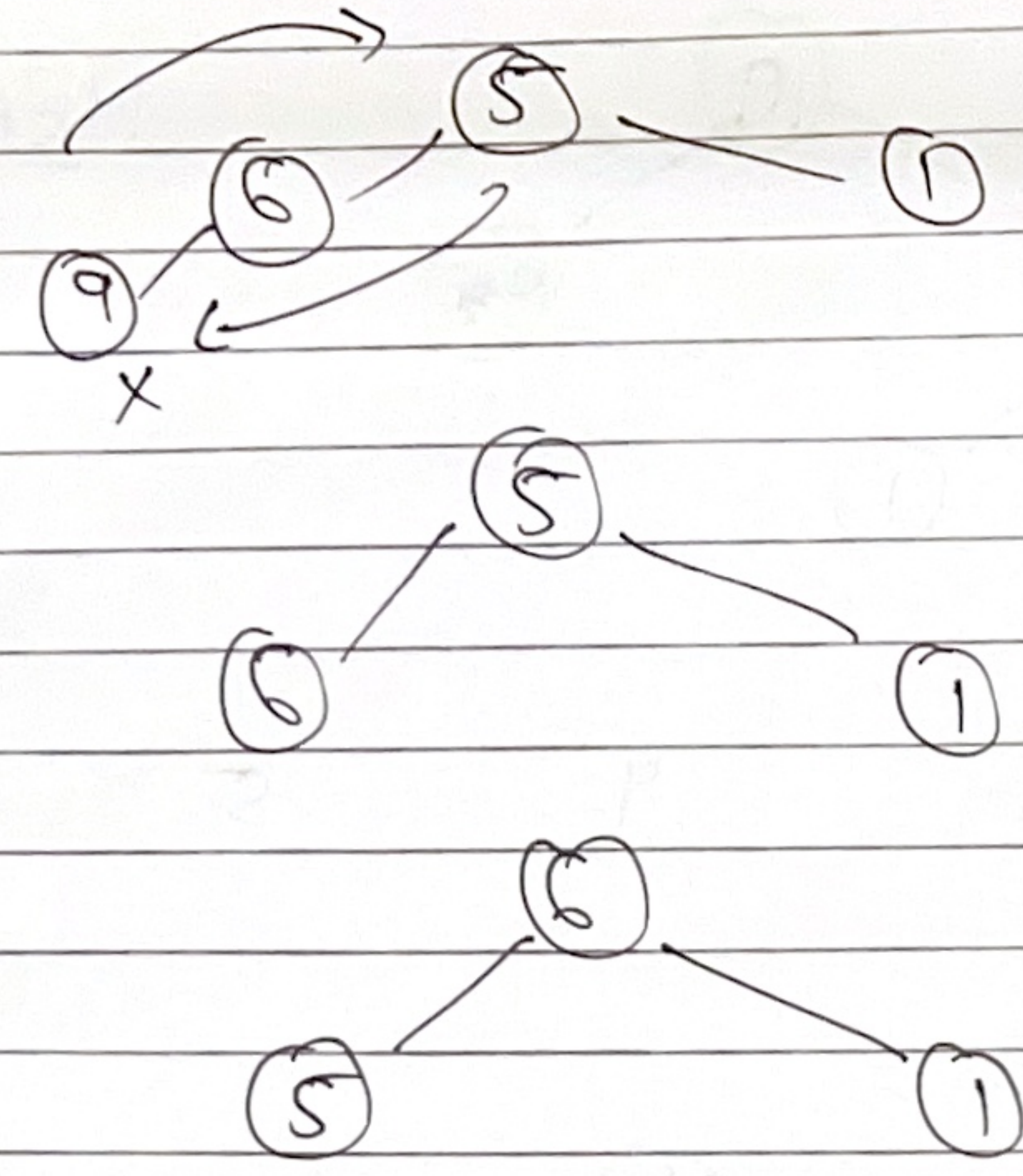
3]



HEAPIFY

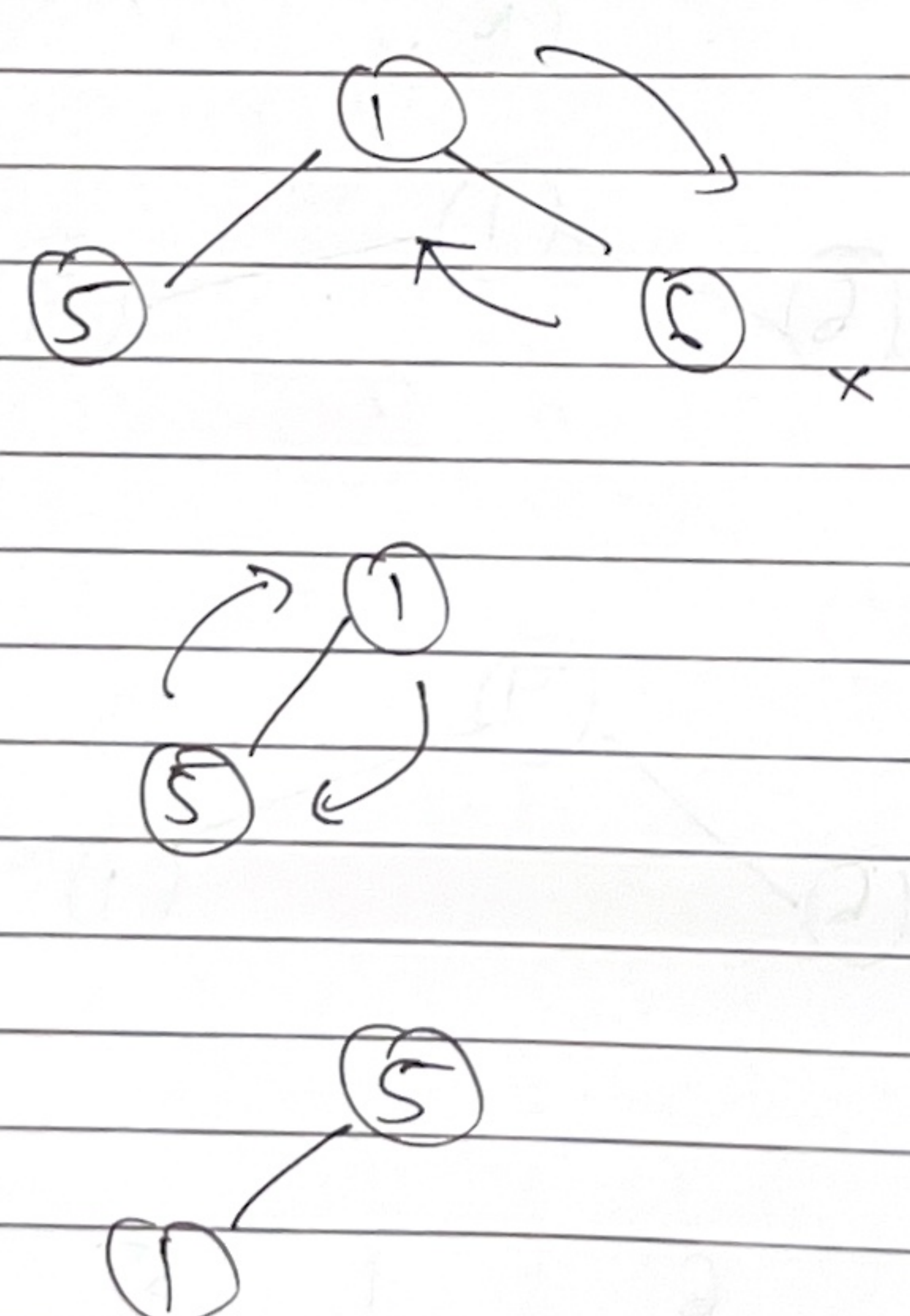
9	6	1	5		
				10	12

PAGE No.
DATE / /



6	5	1			

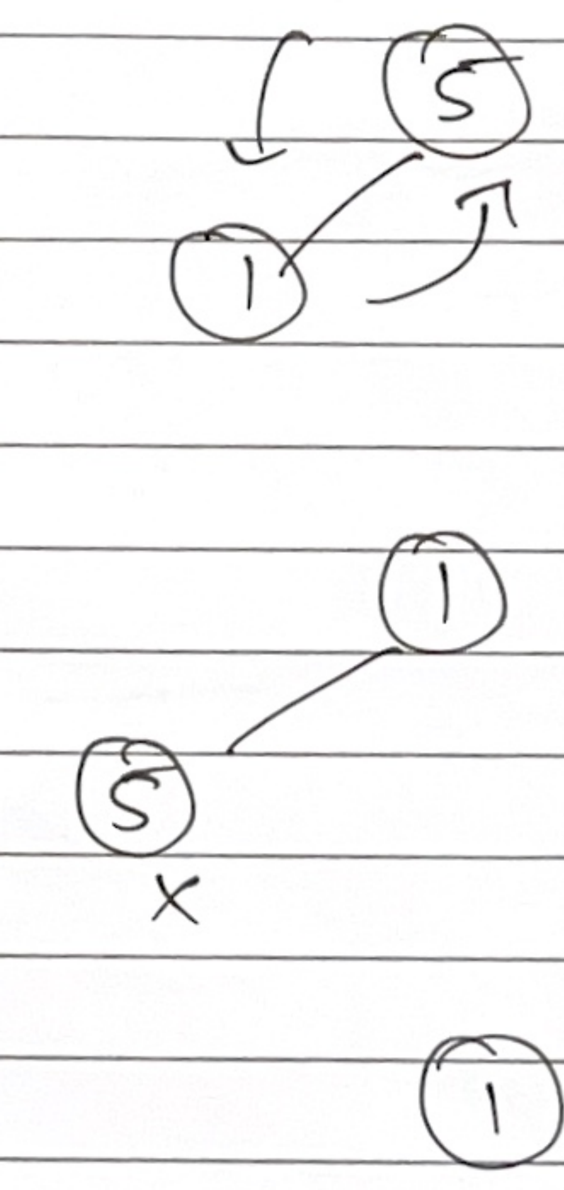
			9	10	12



PAGE No.
DATE / /

5	1				

		6	9	10	12



1					

	5	6	9	10	12

1	5	6	9	10	12

↳ FINAL SORTED ARRAY