GATE

CRASH COURSE

DS & AI

Algorithms

Heap Algorithms
(Lecture 12)

By - Aditya sir

Topics to be Covered

1.

2. Heap Algo:
   - Operations
   - Algos
   - Examples

3. PYQ + Practice

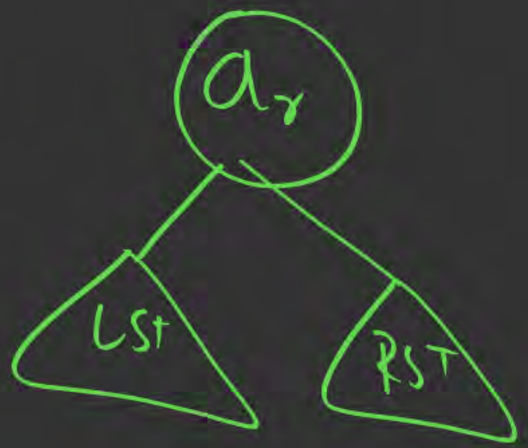
4.

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored working professions in field of Data Science and Analytics
11. Have been mentoring GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

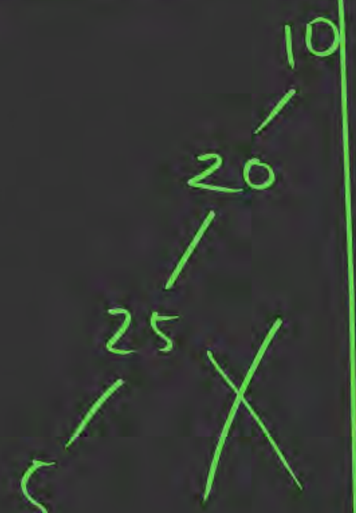
Telegram Link for Aditya Jain sir:

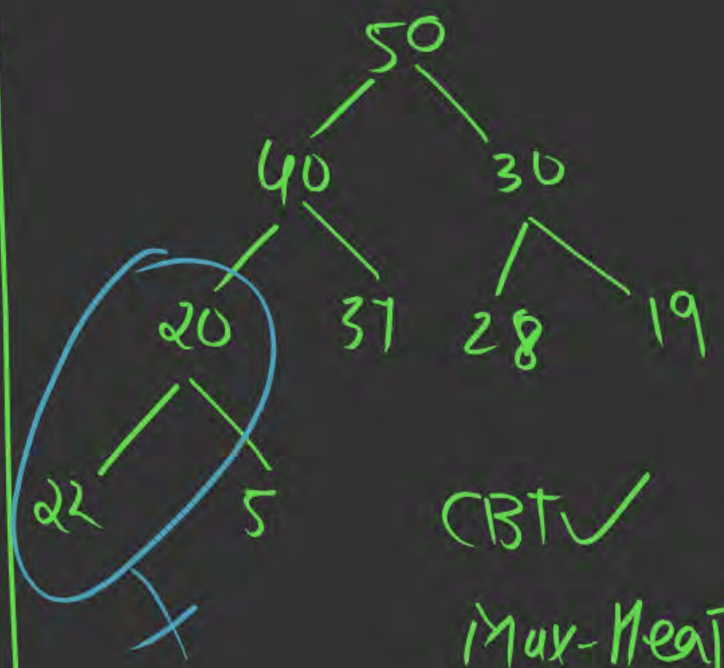
https://t.me/AdityaSir_PW



$a_r \geq (LST, RST)$

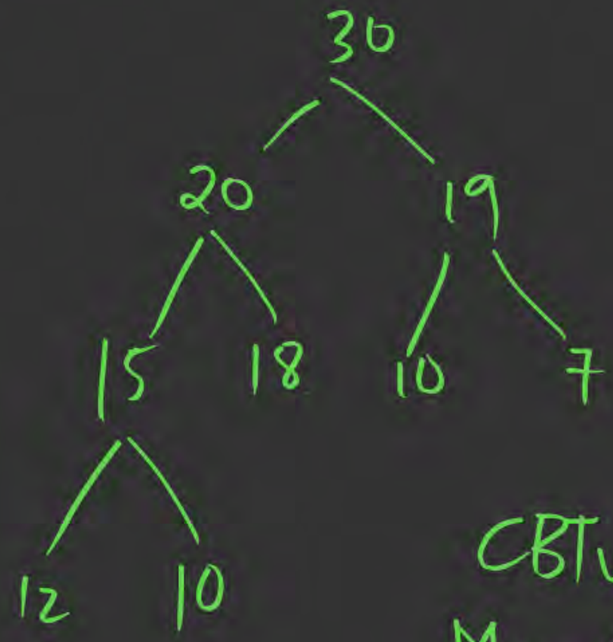
max-Heap





CBT ✓

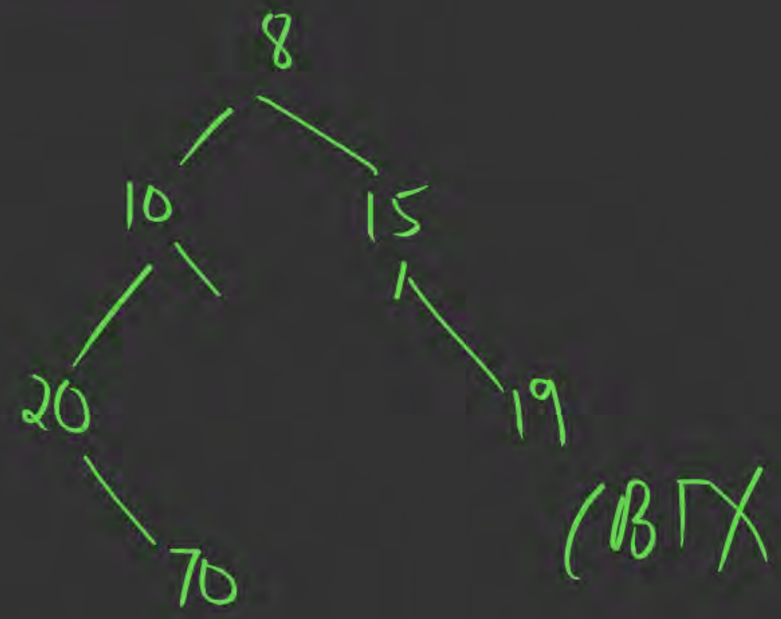
Max-Heap ✗



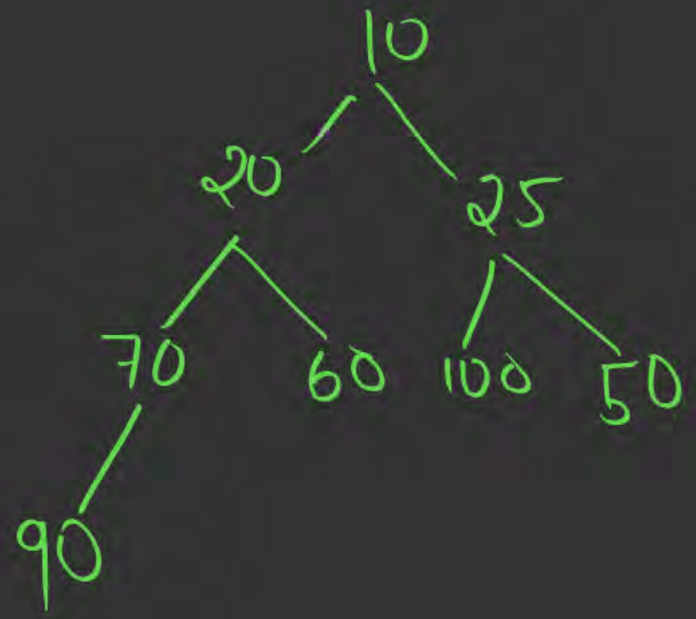
CBT ✓

Max-Heap ✓

Min-Heap



(BST



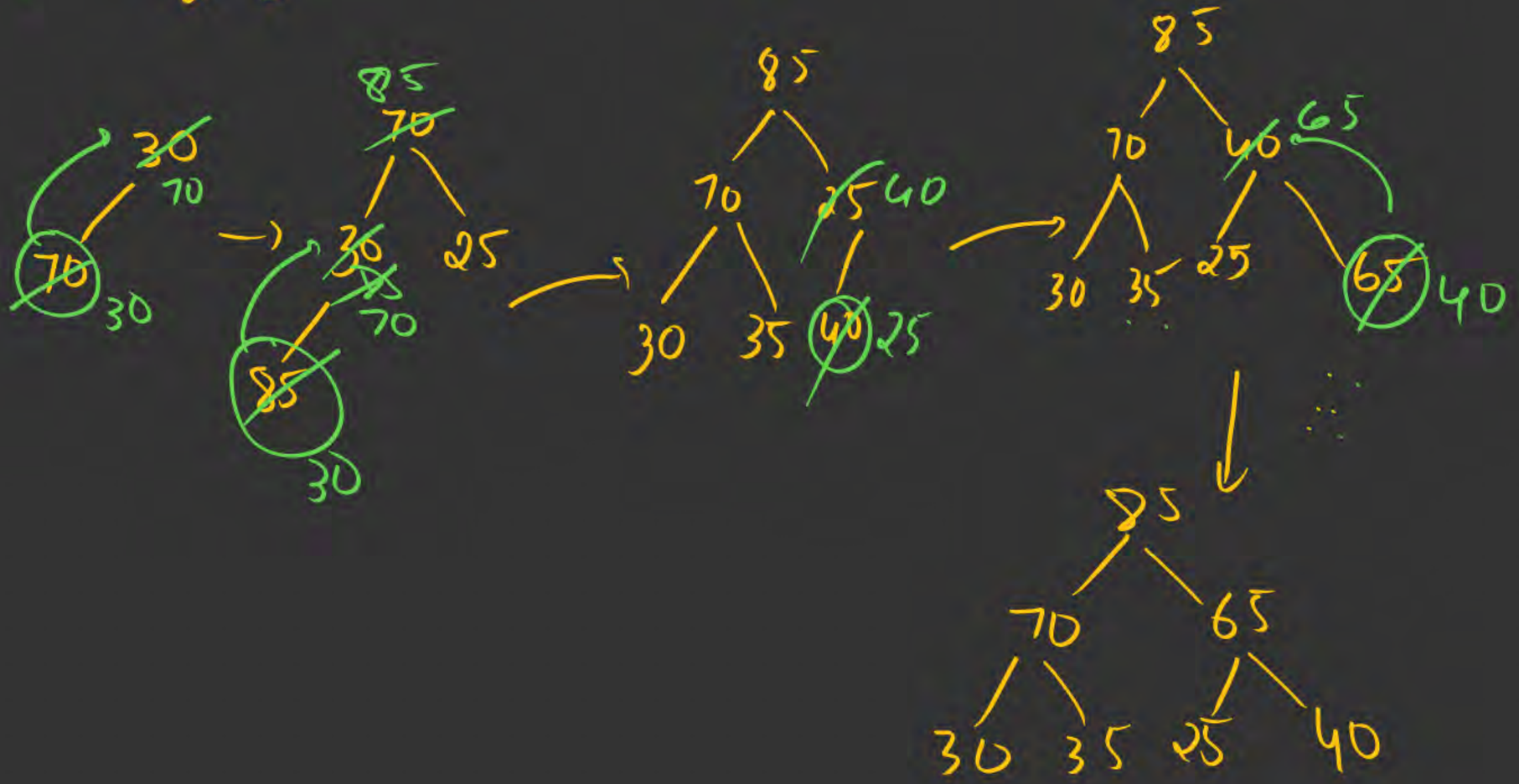
Creation of Heaps :

1) Insertion method
2) Heapify / Build-Heap

Heap → By-default → Max-Heap

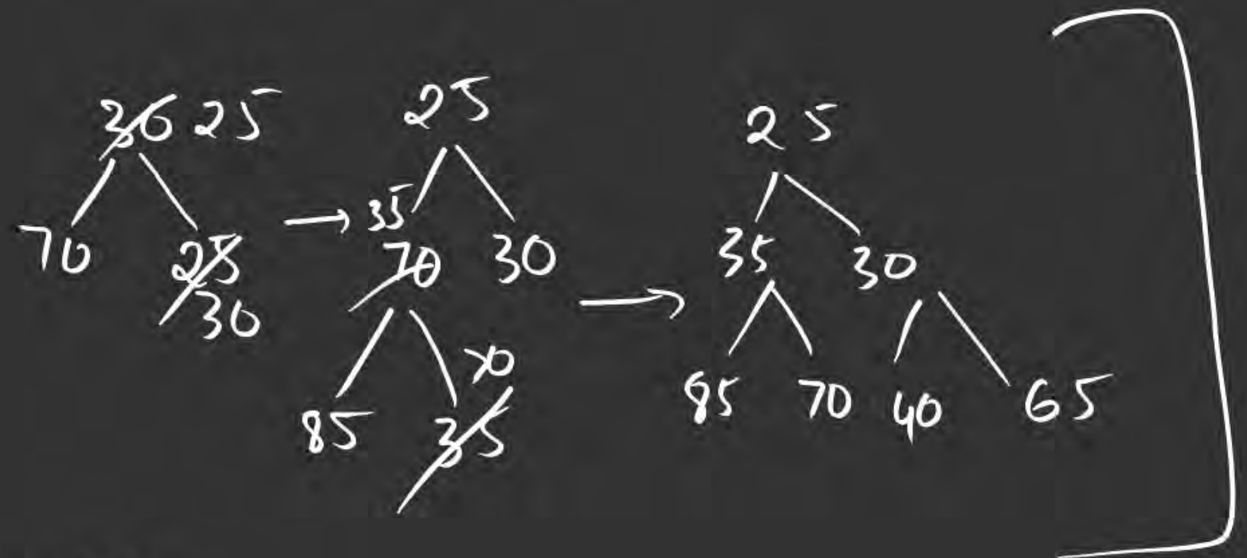
1) Insertion method :

eg:- A = [30, 70, 25, 85, 35, 40, 65]



1) Insertion method :

eg:- A = [30, 70, 25, 85, 35, 40, 65]



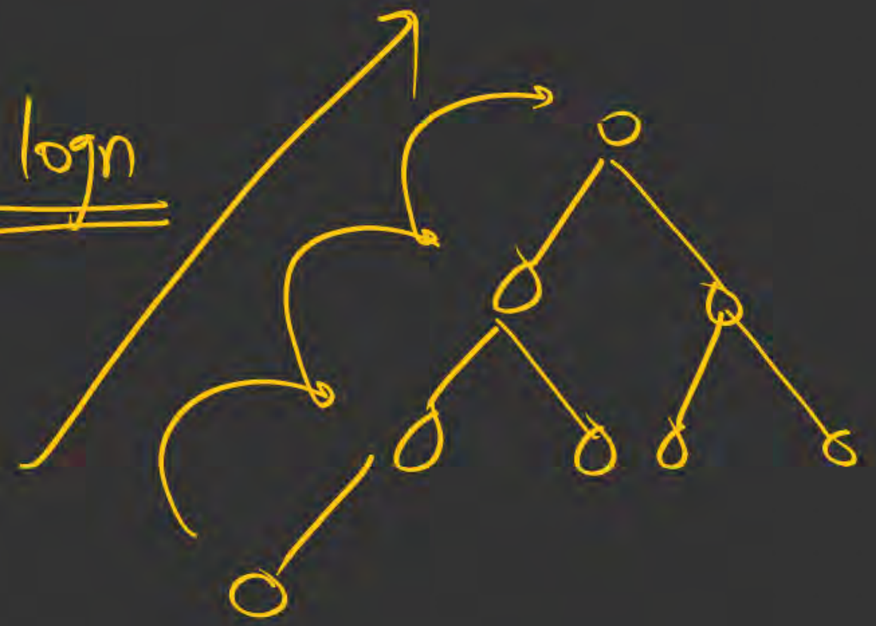
2 swaps

Time Complexity of Insertion operation

Given a Heap of n-elements : insert a node to it.

Best case $\Omega(1)$

Worst case: $O(\log n)$



Heap Creation using Insertion mtd n elems

1) Best Case $n * \Omega(1) \longrightarrow \Omega(n)$

2) Worst Case $n * O(\log_2 n) \longrightarrow O(n \log_2 n)$

$A = [40, 80, 35, 90, 45, 50, 70]$

Heap → Insertion mtd.

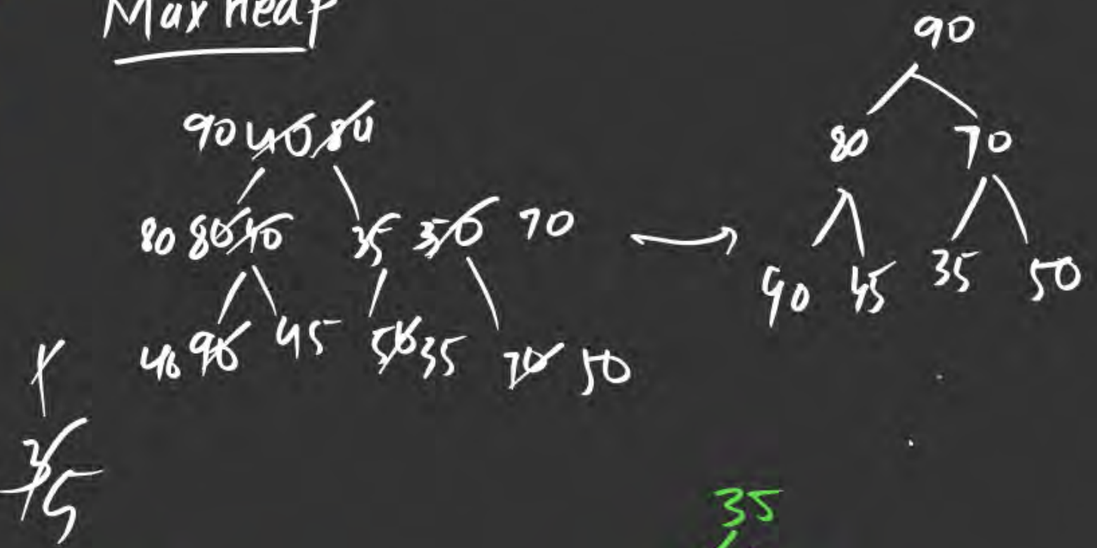
How many sweeps?

Max-Heaps → x

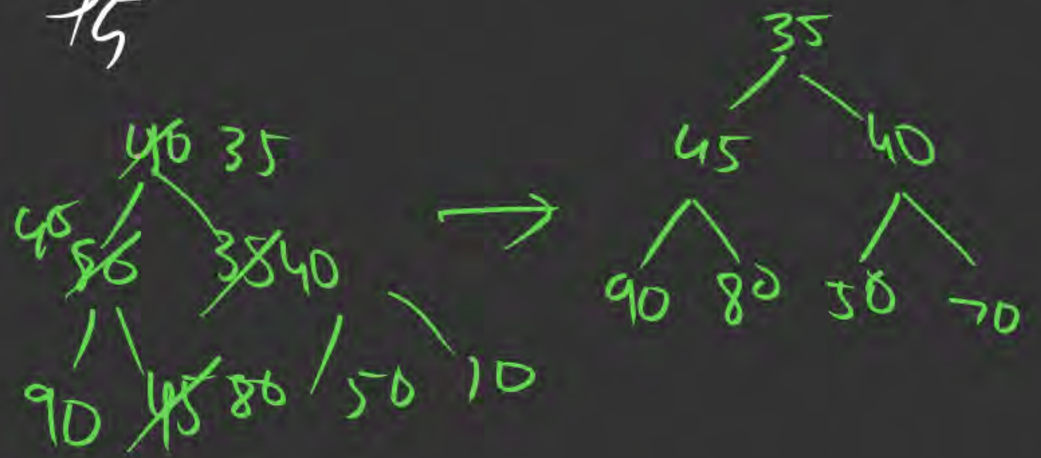
Min-Heaps → y

$x + y = ?$

Max Heap



Min-Heap



$x = 5$

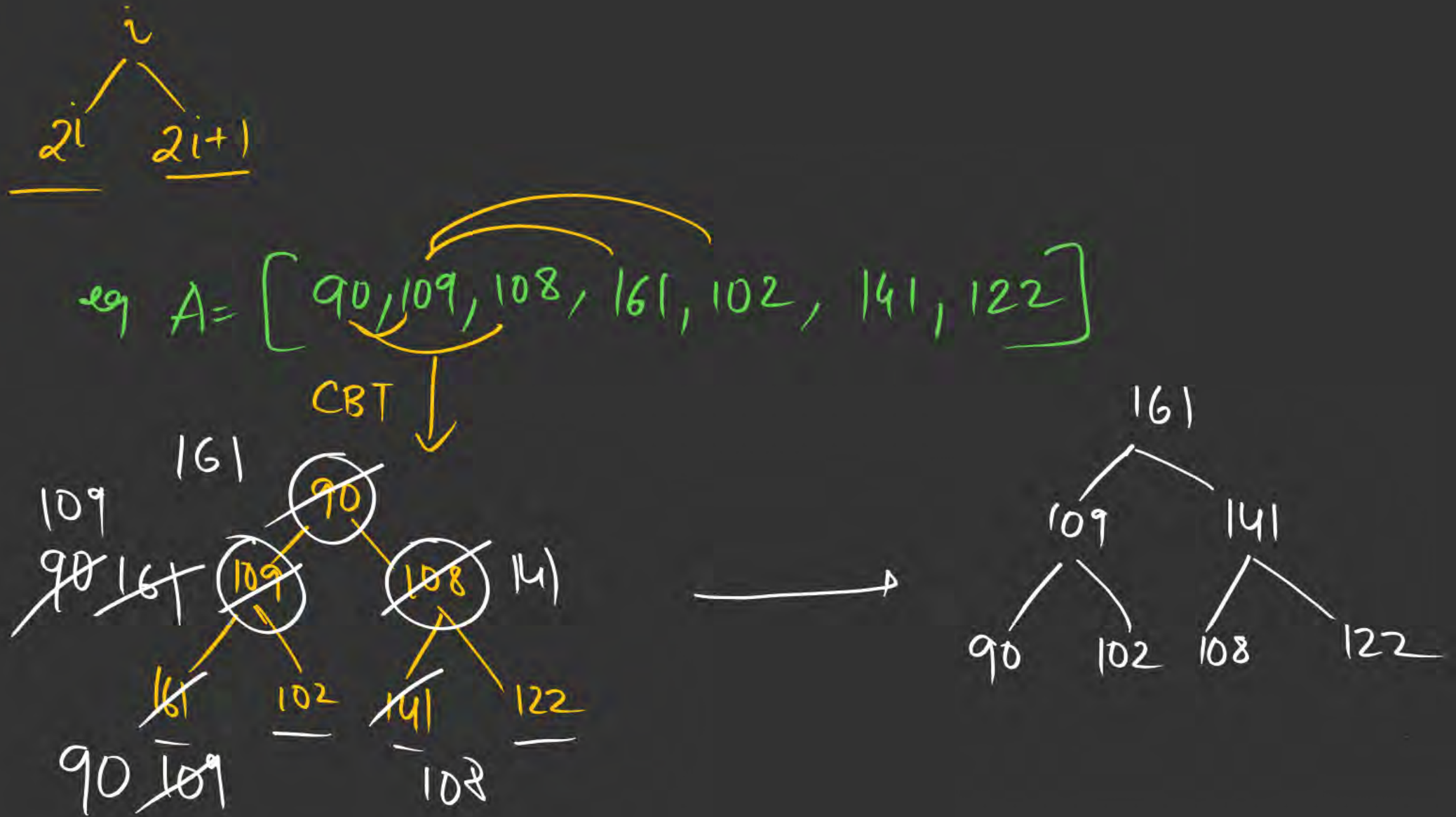
$y = 2$

$x + y = 7$

2) Heapify / Build-Heap:

→ Adjust (level by level)

→ Start at 2nd last level


i
2i
2i+1
eg A= [90, 109, 108, 161, 102, 141, 122]
CBT
161
109
90
109
108
141
161
102
141
122
90
109
108
161
109
141
90
102
108
122

Time Complexity of Heapify :

Heap → n elems

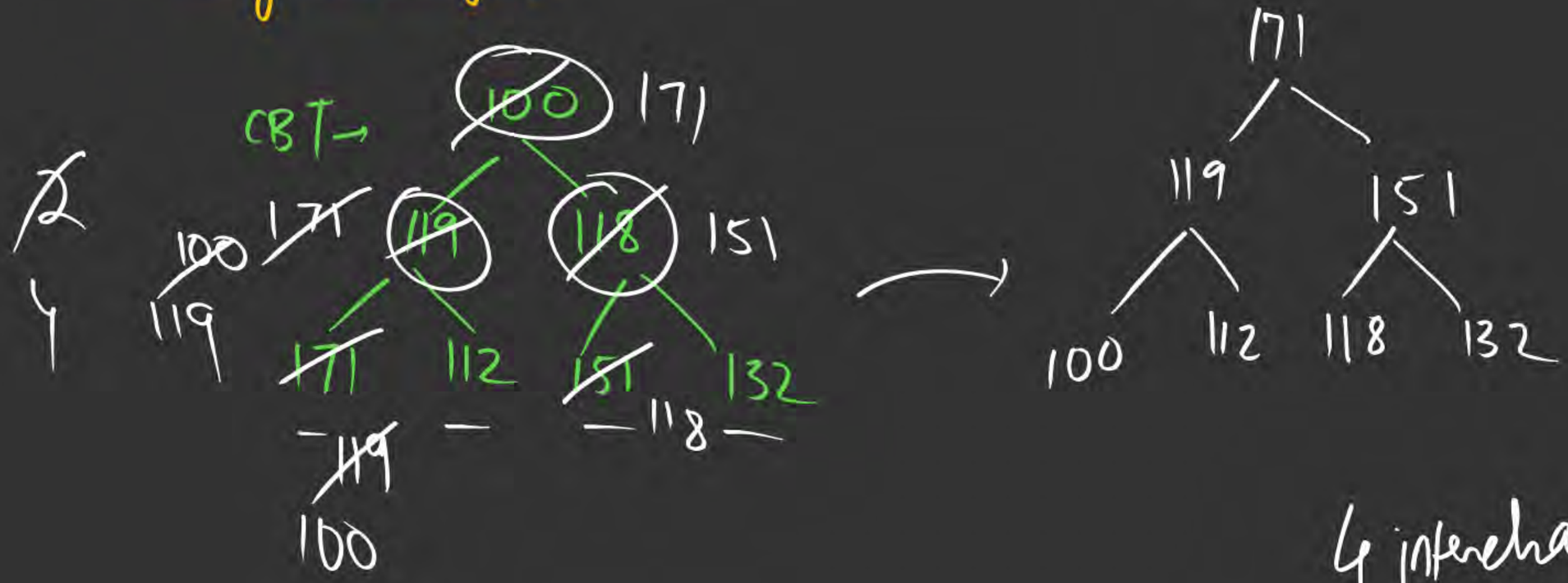
CBT (n) ⟶ Heap

⇒ O(n)

eg: A = [100, 119, 118, 171, 112, 151, 132]

Max-Heap → Heapify

No. of interchanges?



4 interchanges

4) Delete Operation :

Max-Heap ⟶ Delete Root ⟶ max elem

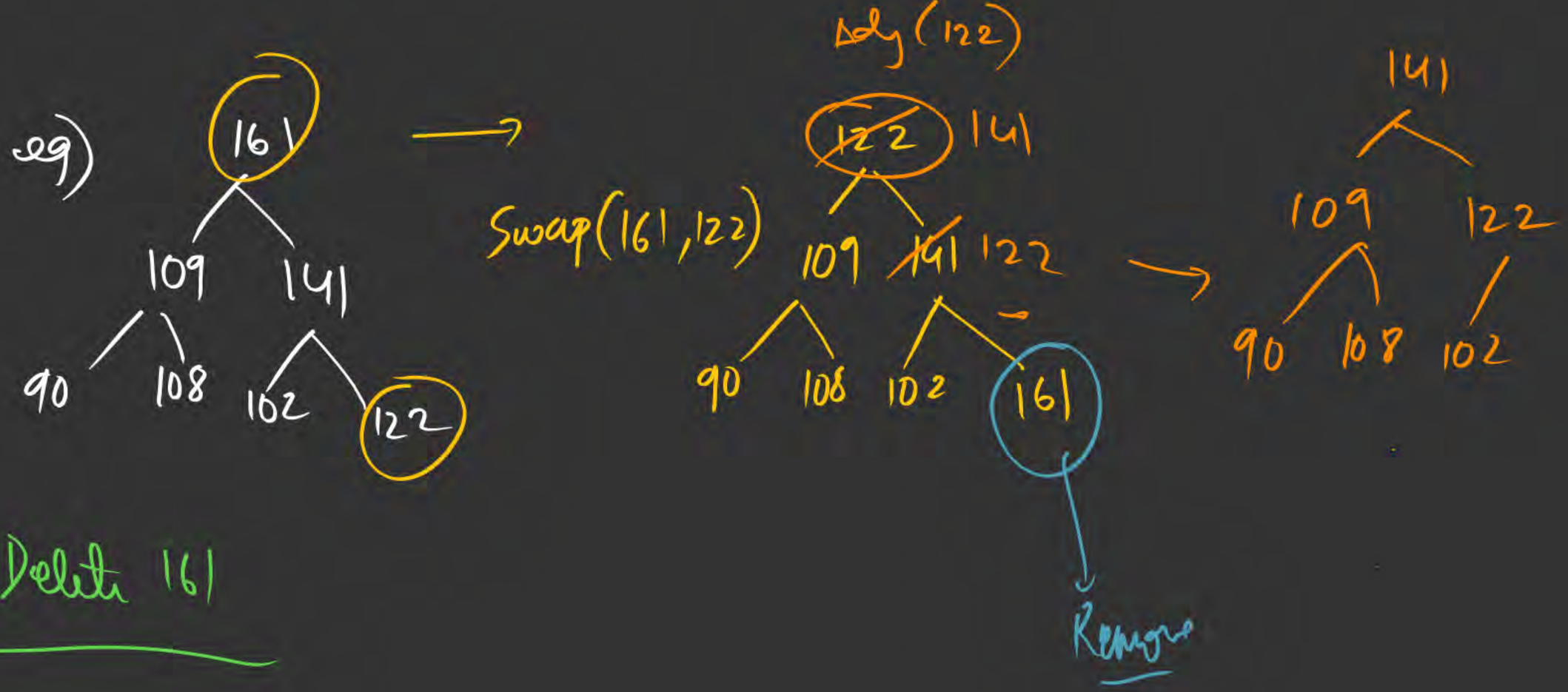
Min Heap ⟶ Delete Root ⟶ min elem

1-based indexing

Steps :- 1) Swap (A[1], A[n])

2) Remove A[n]

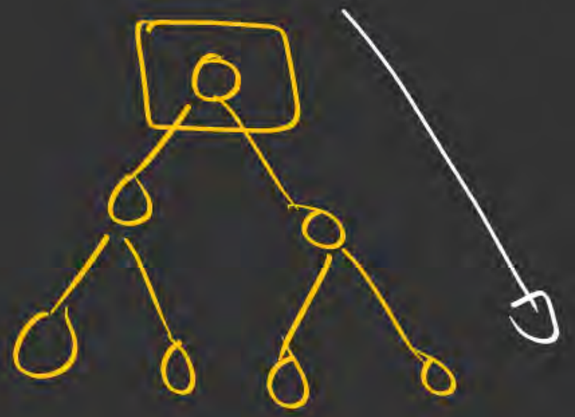
3) Adjust A[1] (Root)


eg)
161
109
141
90
108
102
122
Swap(161,122)
Adj(122)
122
141
109
141
122
90
108
102
161
Remove
141
109
122
90
108
102

Delete 161

Time complexity of delete Operation

Heap (n elems)



Adjust

↳ O(log n)

1 delete op → O(log n)

```
procedure  HEAPSORT (A, n)
// A(1 : n) contains n elements to be sorted.//
1.   call HEAPIFY (A, n)

2.   for i ← n to 2 by – 1 do
        [ t ← A(i); A(i) ← A(1); A(i) ← t
        call ADJUST, (A, 1, i – 1) ]
     repeat
end HEAPSORT
```

I/p Array ⟶ CBT

Heap Sort : n elems

1) Time Complexity → $O(n \log_2 n)$

2) Space Complexity → $O(1)$

3) Inplace

4) Not Stable

Question

#Q.Which Array Representation is a valid Binary Max-Heap

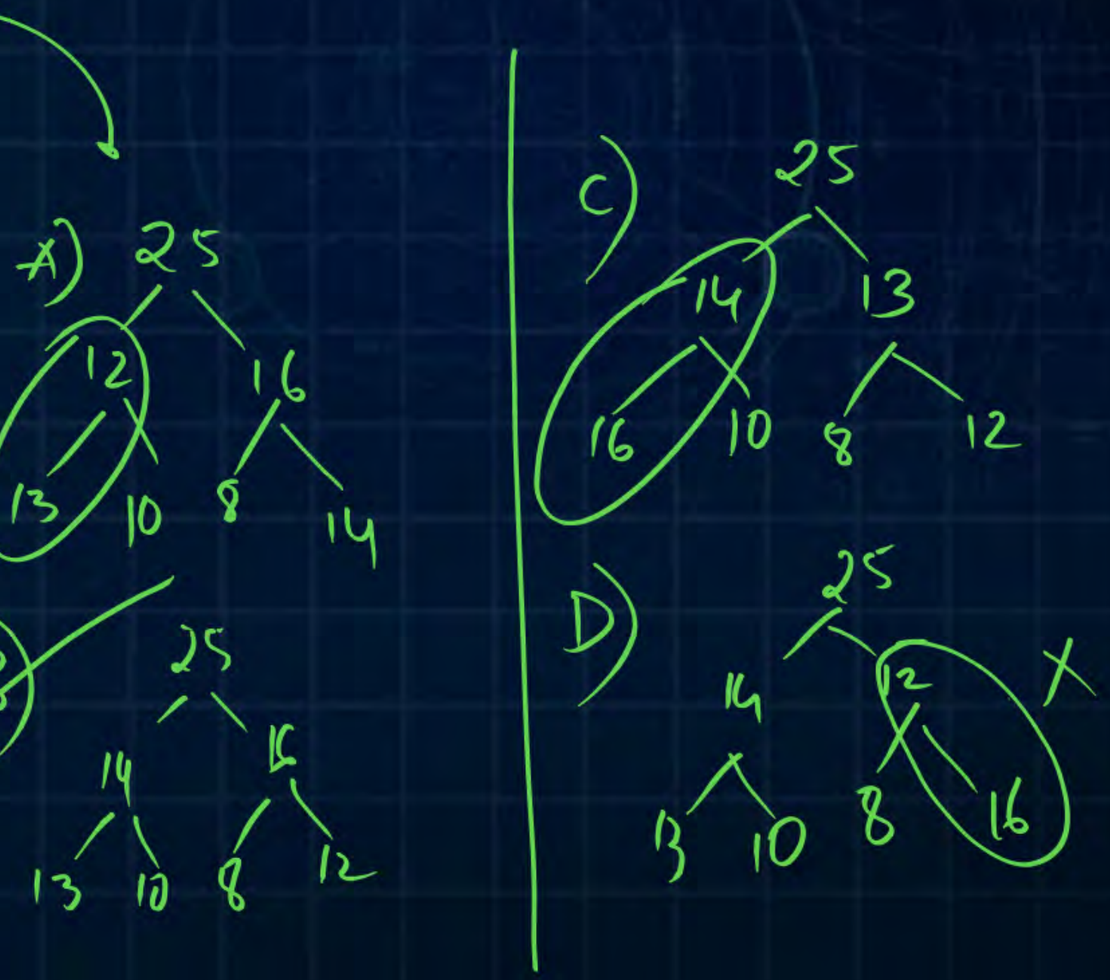
MSQ

A <25, 12, 16, 13, 10, 8, 14>

B <25, 14, 16, 13, 10, 8, 12>

C <25, 14, 13, 16, 10, 8, 12>

D <25, 14, 12, 13, 10, 8, 16>



Question

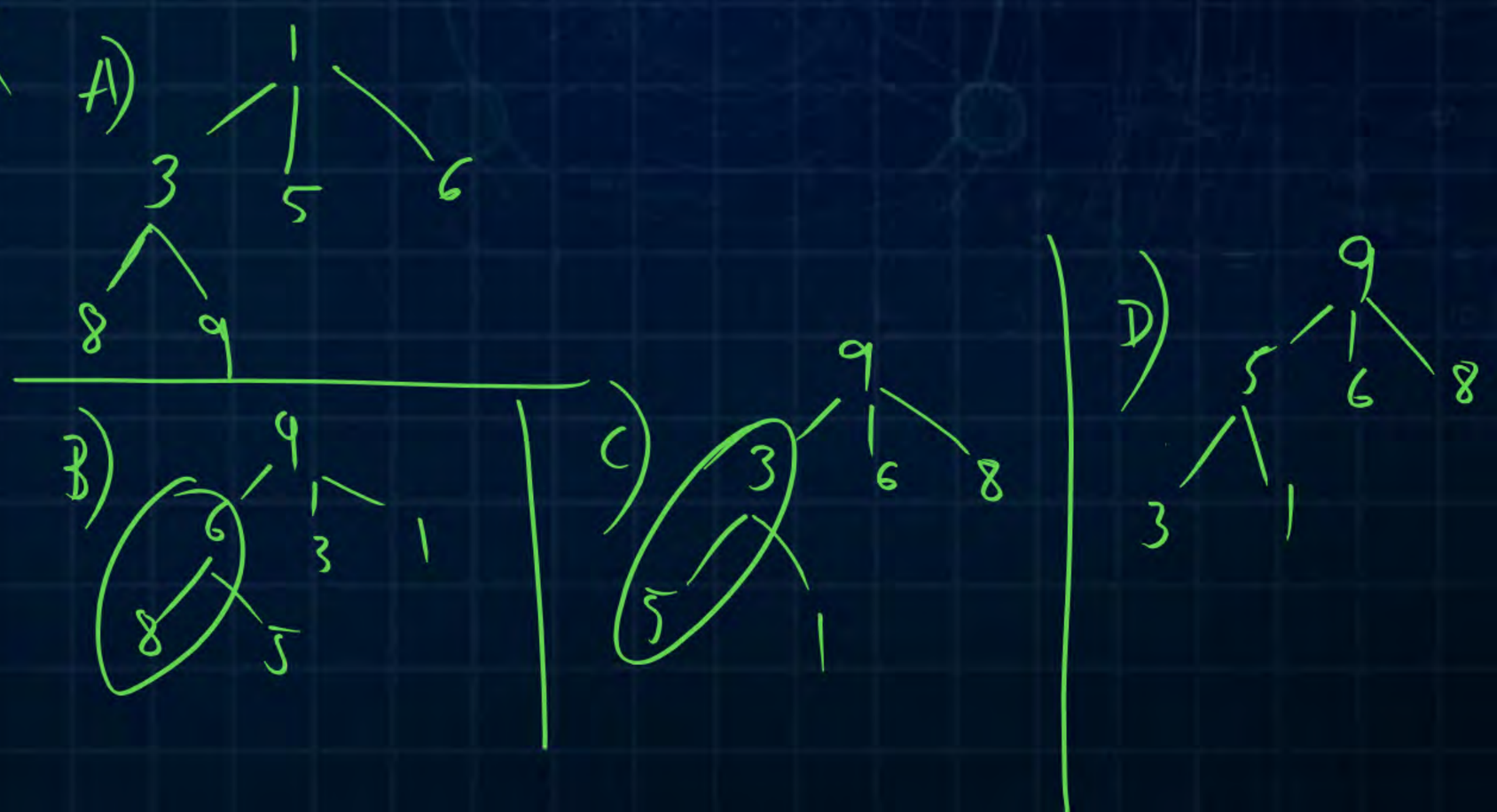
#Q.Which one is valid 3-ary Maximum Heap Array representation

A <1, 3, 5, 6, 8, 9>

B <9, 6, 3, 1, 8, 5>

C <9, 3, 6, 8, 5, 1>

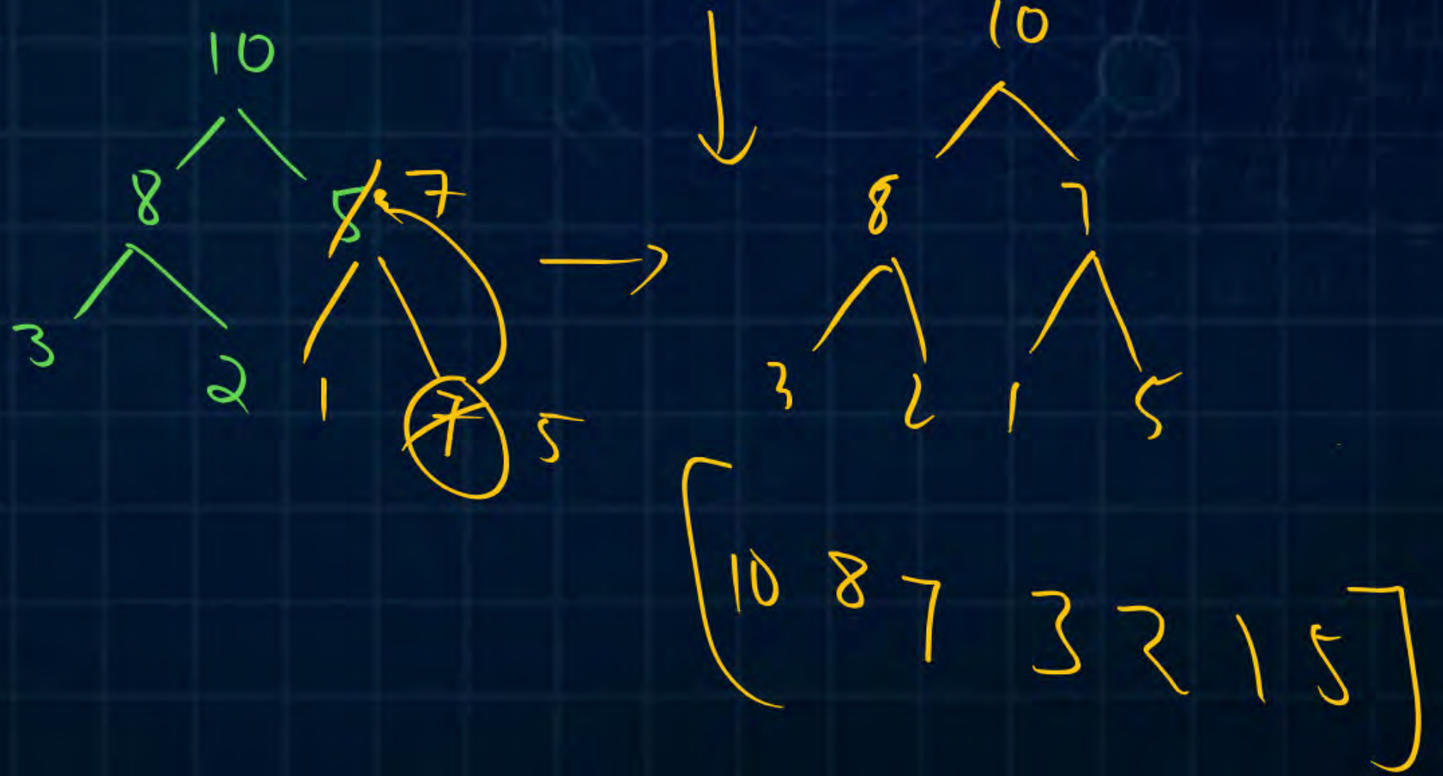
D <9, 5, 6, 8, 3, 1>



Question

#Q. Level order traversal of a binary max heap generates: <10, 8, 5, 3, 2>. To this Heap Insert: <1 and 7>; What is the resultant level order Traversal

given Heap:



Question

#Q. In a Binary Max-Heap with n elements, the smallest element can be found in time of_____.

A) $O(n \log n)$ ✗

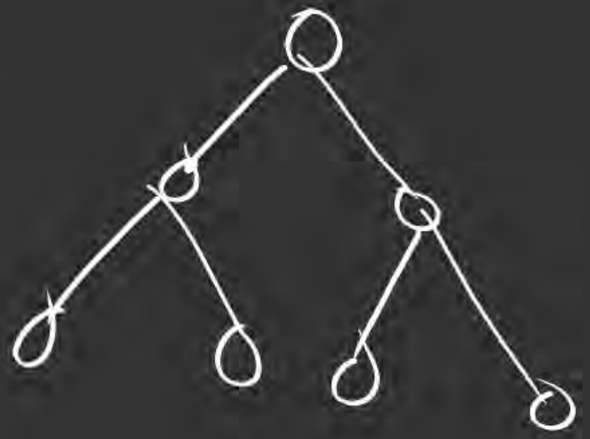
B) $O(\log n)$

C) $O(n)$ ✓

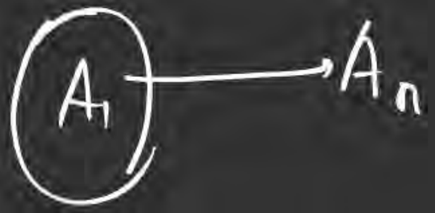
D) $O(n^2)$

" Worst Case Complexity of the most optimal algo "

Case 1

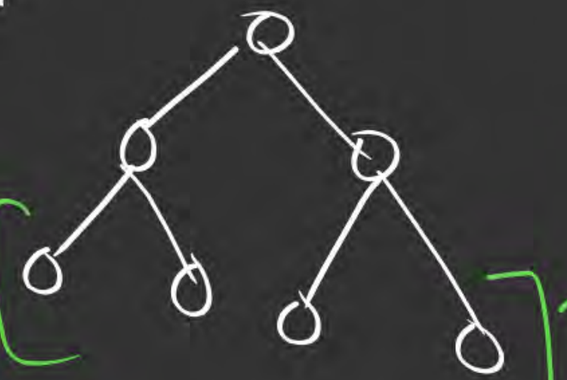


1) Heap → Heap Sort



O(n log n)

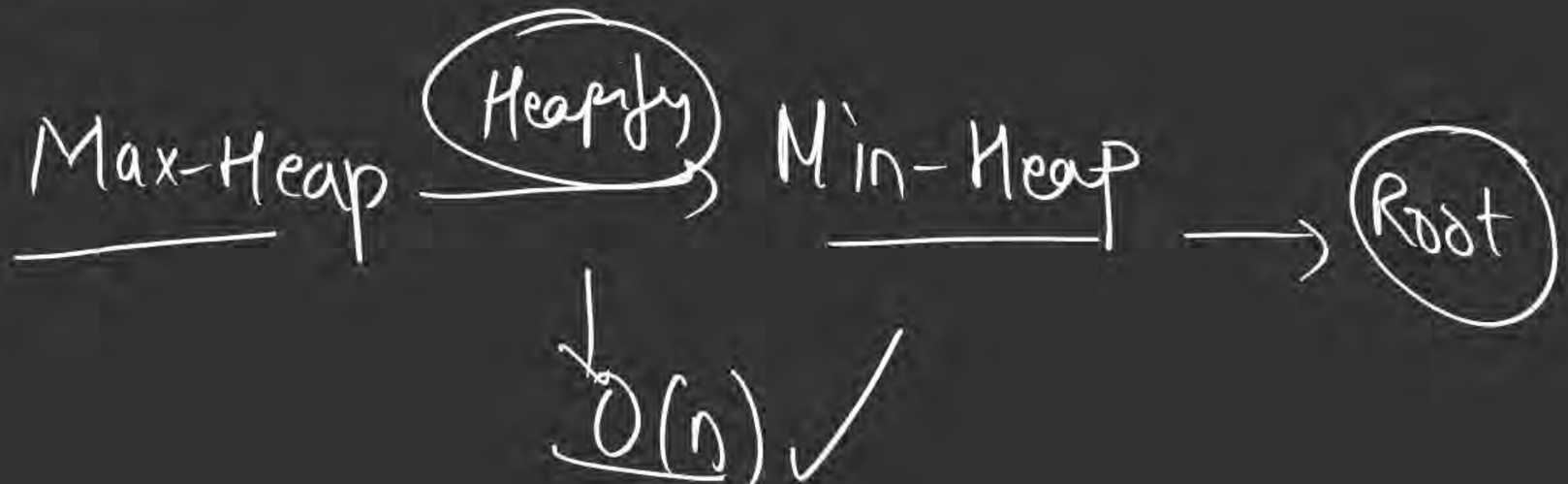
Case2 :-



Max-Heap → min-elem will be at leaf $\approx \lfloor n/2 \rfloor$

$= O(n/2) = O(n)$

Case3:- Heapify



Question

#Q. Given binary Heap with 'n' elements & it is required to insert 'n' more elements not necessarily one after another into this Heap. Total time required for this operation is:

A $O(n^2)$

B nlogn

C n

D n^2logn

Given

n new elems

appr-1

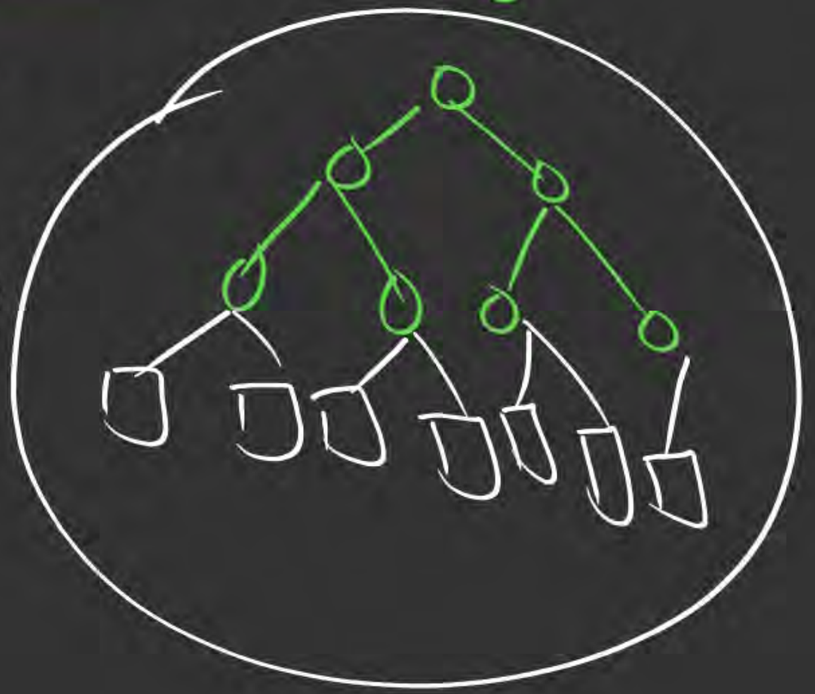
i) Insertion into heap

every insert → O(log n)

→ O(n log n)

Approach-2 : Heapify / Build Heap

CBT of 2n elements



Heapify

CBT ⟶ MaxHeap

(2n) (2n)

Heapify → O(2n) → O(n)

Question

#Q. The approximate number of element that can be sorted in O(log n) time using Heap Sort is _____.

A) $O(\log n)$ X

B) $O(1)$ X

C) $O(n)$ X

D) $O\left(\frac{\log n}{\log(\log n)}\right)$

Soln : x elems $\longrightarrow$ $O(x \log_2(x))$

A) $\log n$ elems $\longrightarrow$ $O(\log n * \log(\log n))$ ✗

B) $O(1)$ elms ✗

C) $O(n)$ elems $\longrightarrow$ $O(n \log n)$ ✗

1) n elems $\longrightarrow O(n \log_2 n)$

$\frac{\log n}{\log(\log n)}$ elems $\rightarrow O\left(\frac{\log n}{\log(\log n)} * \log\left(\frac{\log n}{\log(\log n)}\right)\right)$

$\log(A/B) = \log A - \log B$

$= O\left(\frac{\log n}{\log(\log n)} * \left[\log\log n - \log\log\log n\right]\right)$

$= O\left(\log n - \frac{\log n}{\log\log n} * \log\log\log n\right)$

$\frac{\log n}{\log\log n} * \log\log\log n \leq 1$

$= O(\log n)$

Question

#Q. Consider a max heap, represented by the array:

40, 30, 20, 10, 15, 16, 17, 8, 4

Array index	1	2	3	4	5	6	7	8	9
Value	40	30	20	10	15	16	17	8	4

Now consider that a value 35 is inserted into this heap. After insertion, the new heap ~~hope~~ is

A 40, 30, 20, 10, 15, 16, 17, 8, 4, 35

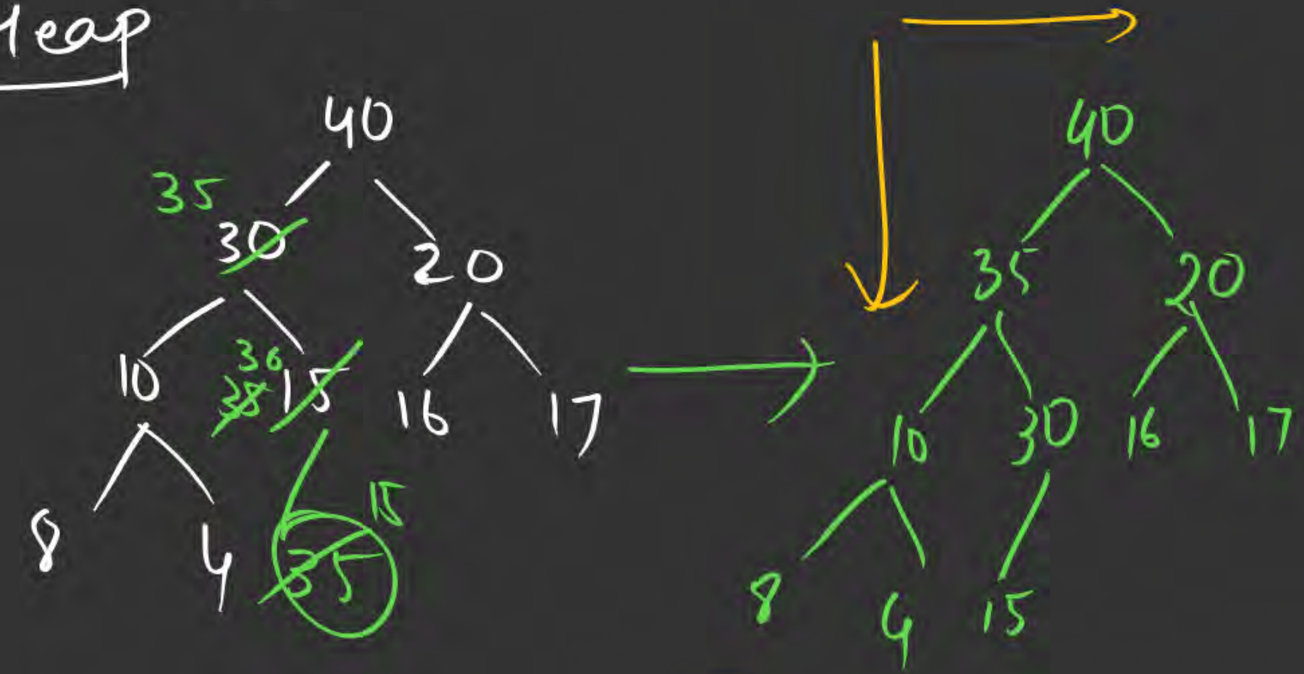
B 40, 35, 20, 10, 30, 16, 17, 8, 4, 15

C 40, 30, 20, 10, 35, 16, 17, 8, 4, 15

D 40, 35, 20, 10, 15, 16, 17, 8, 4, 30

Solnt A: 40, 30, ~~20~~ 20, 10, 15, 16, 17, 8, 4

Given Heap



O/p: [40, 35, 20, 10, 30, 16, 17, 8, 4, 15]

Question

of n elems

#Q. Given Binary Heap in Array with the smallest at the root, the 7th smallest element can be found in time complexity of ____.

min-Heap

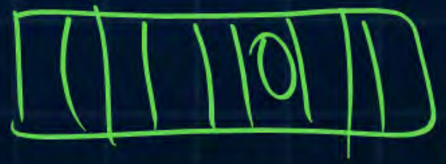
A) O(n)

B) O(logn)

C) O(1)

D) O(nlogn)

Heap Sort



Soln :- Traditional

1 deletion ⟶ 1st min

2 deletion ⟶ 2nd min

3 " ⟶ 3rd min

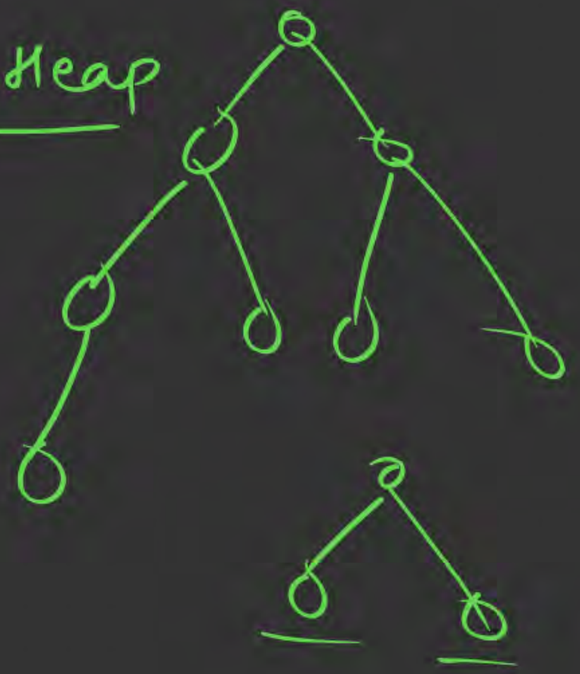
⋮

7th deletion ⟶ 7th min

1 deletion ⟶ $O(\log_2 n)$

↳ 7 deletion ⟶ $O(7 * \log_2 n) = O(\log n)$

min-Heap

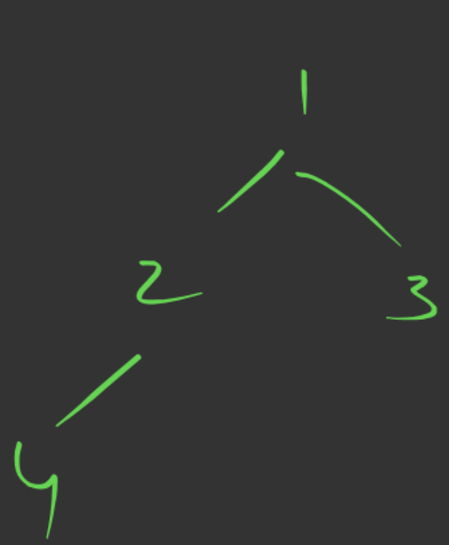


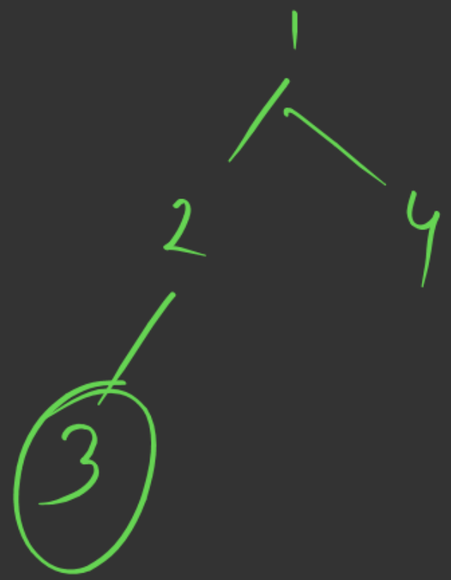
Max-depth

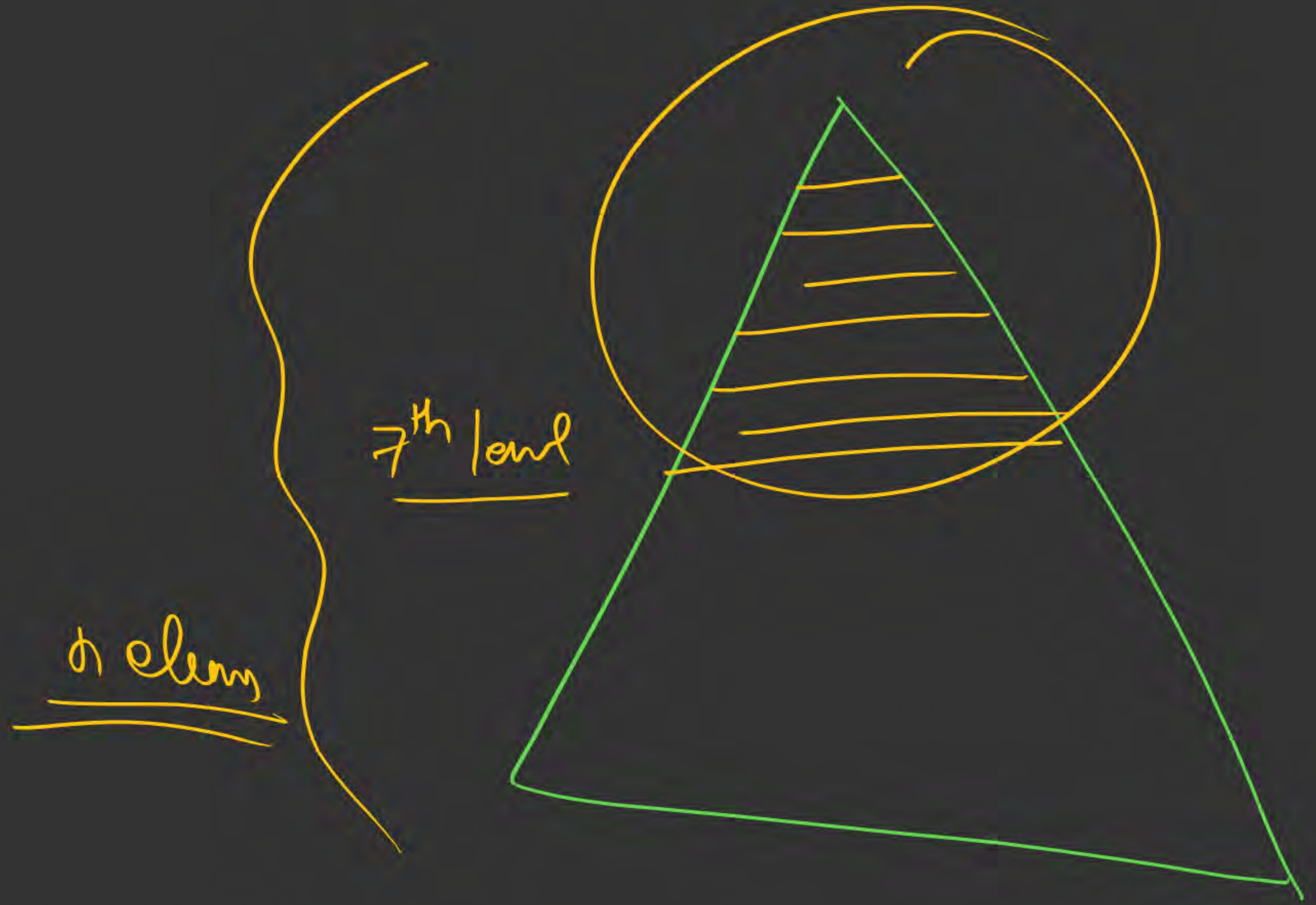
1st min elem ⟶ 1st level

2nd min elem ⟶ 2nd level

3rd min elem ⟶ ?


1
2
3
4


1
2
4
3


7th level
n elems

1st level ⟶ 2^0 = 1 elem.

2nd level ⟶ 2^1 = 2 elems

3rd level ⟶ 2^2 = 4 elems

4th level ⟶ 2^3 = 8 elems

i^{th} level ⟶ $2^{(i-1)}$ elems

Total elems within Top 7 levels.

$= (2^0 + 2^1 + \cdots 2^7)$

$\approx (2^7 - 1)$; independent of n

Thank

THANK

Keep Hustling!