

COL106: Data Structures and Algorithms

Lecture 28

Keerti Choudhary
Department of Computer Science, IIT Delhi

Q1. Merging 2 BSTs of $O(n)$ size.

$BST_1 \rightarrow$ sorted array A_1

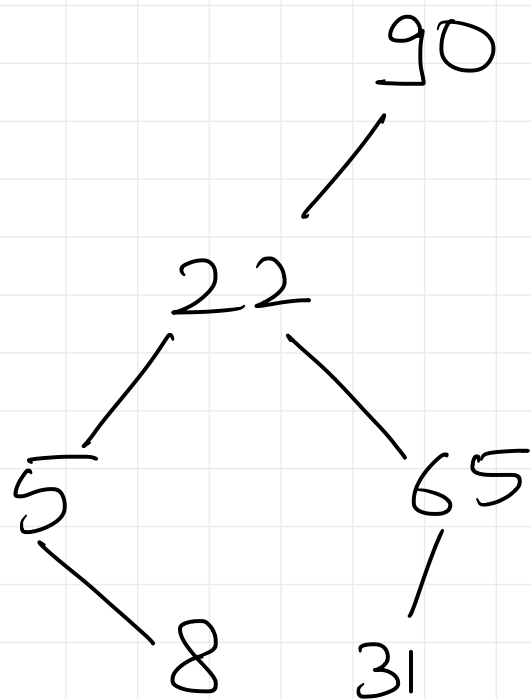
$BST_2 \rightarrow$ sorted array A_2

Merge A_1, A_2 in $O(n)$ time to get sorted array B .

Convert B to a BST (balanced?) in $O(n)$ time.

Q2

BST $\leftarrow$ 90, 22, 5, 65, 8, 31



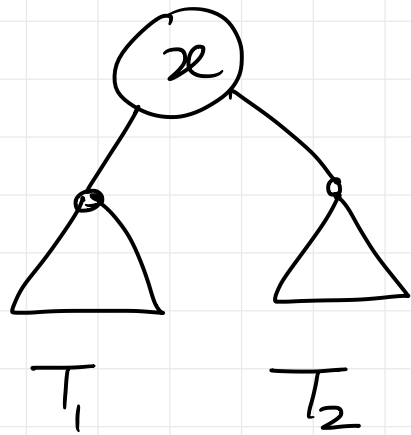
Inorder : 5 8 22 31 65 90

Preorder : 90 22 5 8 65 31

Postorder : H.W.

Q3

Let $\mathcal{N}(h)$ = Min no. of nodes in AVL Tree of height h



Let $h = \text{height}(T)$

So, $\left. \begin{array}{l} \text{height}(T_1) = h-1 \\ \text{height}(T_2) = h-2 \end{array} \right\}$ for min no of nodes.

$$\Rightarrow \mathcal{N}(h) = 1 + \mathcal{N}(h-1) + \mathcal{N}(h-2)$$

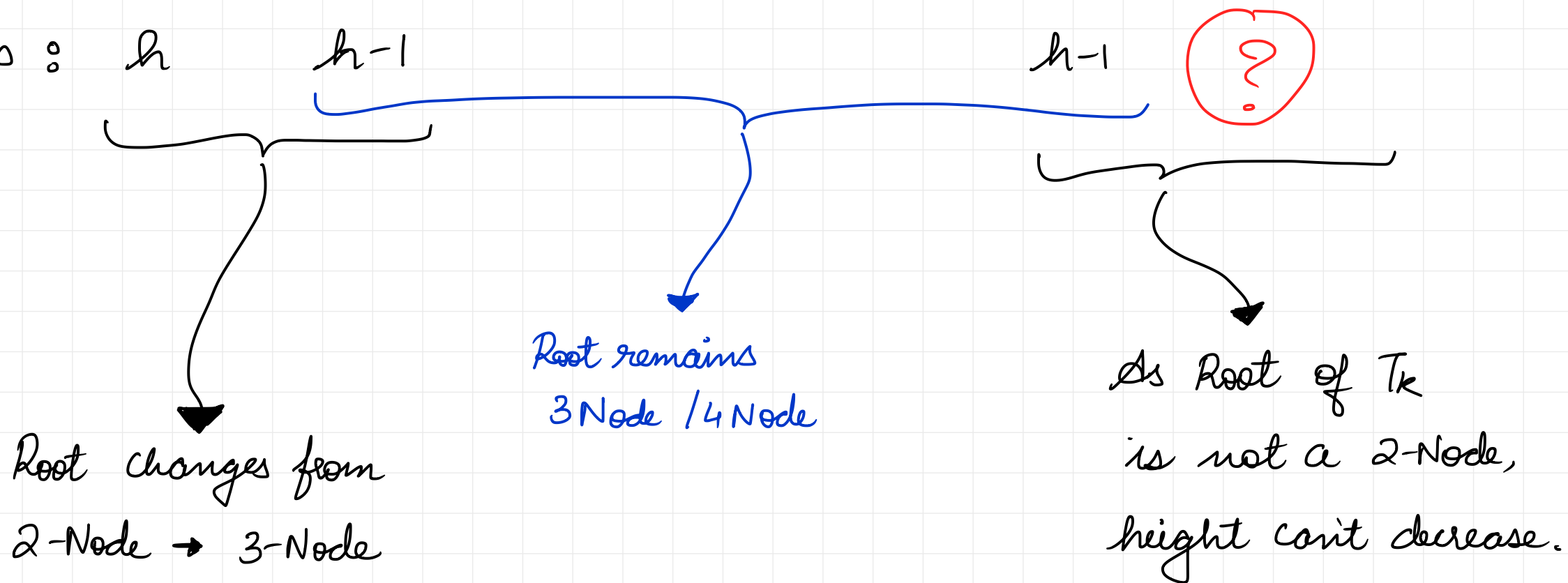
h	0	1	2	3	4	5	6	7	8
$\mathcal{N}(h)$	1	2	4	7	12	20	33	54	88

Q4

T_0 : 234 tree

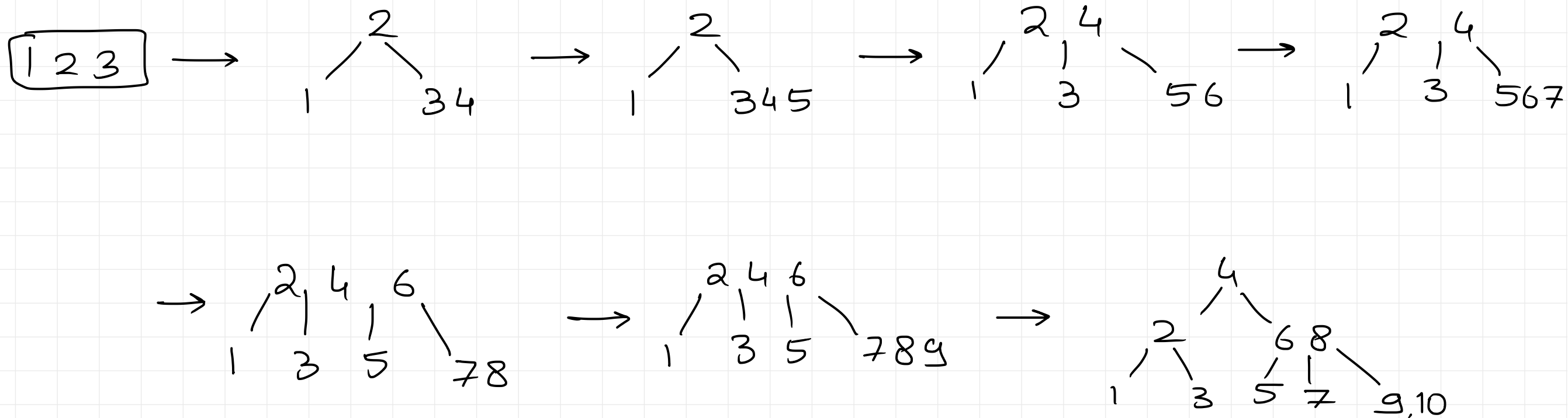
Updates to Tree : $T_0 \xrightarrow{+} T_1 \xrightarrow{-} T_2 \xrightarrow{-} \dots \xrightarrow{-} T_{k-1} \xrightarrow{-} T_k \xrightarrow{+} T_{k+1}$

Modifications :
in height



Q5

Seq: 1, 2, 3, 4, 5, 6, 7, ..., 17, 18



Q6

Merging 2 Heaps

Convert heap H_1 to array A_1
———"——" H_2 ———"——" A_2

B = Array obtained by merging A_1 and A_2

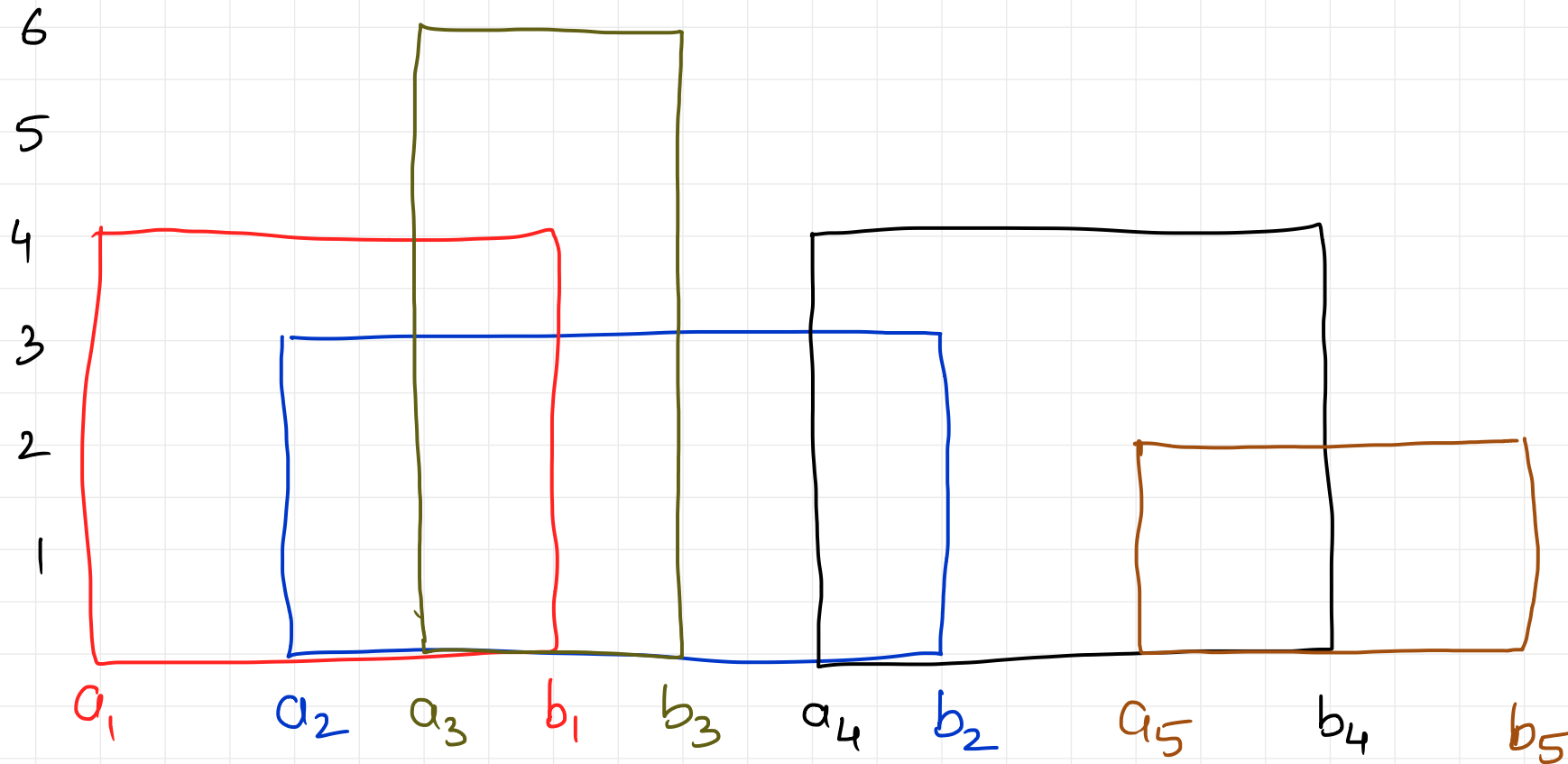
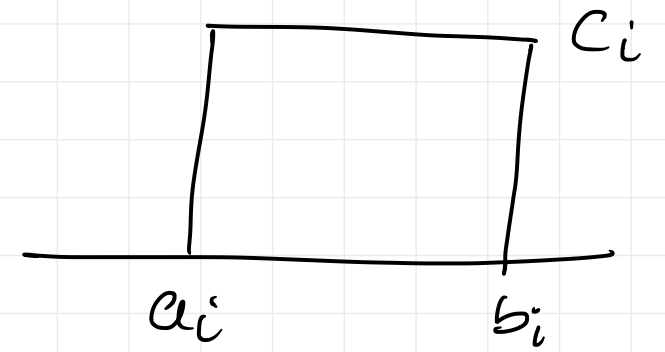
Call Heapify (B).

$$\text{Time} = O(\text{size}(B)) = O(m+n)$$

H.W. : Time to make Heap on an n -sized array is $O(n)$.

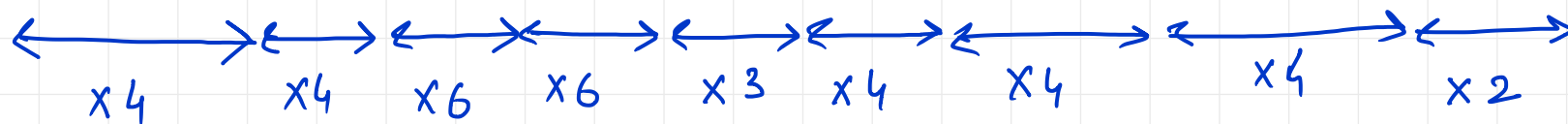
Q7 $X =$ Sorted sequence of a_i 's and b_i 's

Assumption: All a_i 's and b_i 's are distinct



Time = $O(n \log n)$

Heap 4 34 346 36 3 34 4 24 2 $\emptyset$



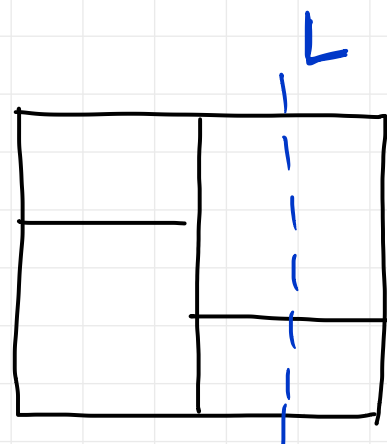
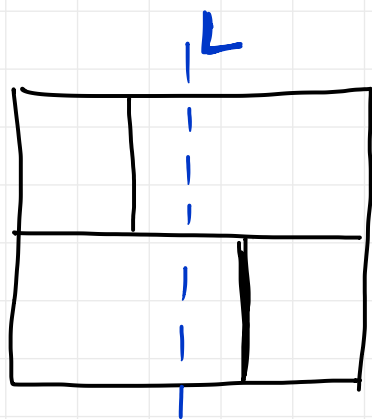
H.W. : How to insert/delete arbitrary entry of max heap in $O(\log n)$ time.

Q8 Axis-Parallel Range Query.

Subproblem : Given an axis parallel line segment L (| or —), how many nodes in 2d tree can intersect / contain it?

Solution

$Q(n) = 2 + 2Q(n/4)$, because when a node is divided into four parts of size $n/4$, then exactly 2 out of 4 subnodes can intersect L .



Solving this gives $Q(n) \leq 4\sqrt{n} - 2$

Now, for an axis parallel rectangle, let

$A =$ All nodes of 2d tree intersecting / containing rectangle edges.

$B =$ All nodes x of 2d tree lying completely inside 2d tree.

$C =$ All leaf node x of 2d tree lying completely inside 2d tree.

$$\# \text{ of } A = O(\sqrt{n})$$

$$\# \text{ of } C = O(m)$$

$$\# \text{ of } B = O(m) \text{ because } B = O(C).$$

$$\text{So, ans} = O(\#A + \#B + \#C) = O(m + \sqrt{n})$$

Q9

Compute $\text{dist}(s, x)$ and $\text{dist}(x, t)$ for each $x \in V(G)$.

This takes $O(m+n)$ time as we simply need to compute $\text{BFS}(s)$.

Initialize $S = \emptyset$

For each (x, y) ,

add (x, y) to S iff $\text{dist}(s, t) = \text{dist}(s, x) + 1 + \text{dist}(y, t)$

Return S

Q10. Graph Coloring

G is 2-colorable $\Leftrightarrow G$ is bipartite $\Leftrightarrow G$ has no odd length cycles.