

# **COL 351:**

# **Analysis and Design of Algorithms**

## **Lecture 12**

# Longest Increasing Subsequence (LIS)

$$A = [8 \ 5 \ 7 \ 3 \ 9 \ 4 \ 11 \ 2]$$

$$\text{LIS}(A) = [5 \ 7 \ 9 \ 11]$$

Find a longest subsequence  
of  $A$  which is strictly increasing

Assumption — All elements of  $A$  are distinct

# Subproblem to solve LIS?

Input: An integer array  $A$  of size  $n$

Subproblem (attempt 1):

$$L[i] := \text{length of LIS of } A[1, i]$$

$$A = [8 \ 5 \ 7 \ 3 \ 9 \ 4 \ 11 \ 2]$$

$$L = [1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 4 \ 4]$$

We can't compute  $L[i]$  from  $L[1], \dots, L[i-1]$

# Subproblem to solve LIS?

*Input:* An integer array  $A$  of size  $n$

*Subproblem (attempt 2):*

$L[i] := \text{Length of LIS of } A[1, i] \text{ ending at } A[i].$

# Solving LIS

**Input:** An integer array  $A$  of size  $n$

**Subproblem:**

$L[i] :=$  length of LIS of  $A[1, i]$  ending at  $A[i]$ .

**claim:**

$$L[i] = 1 + \max \{ L[j] \mid j < i, A[j] < A[i] \}$$

**Proof:**

H.W.

---

**Theorem:** We can compute LIS in  $O(n^2)$  time.

# Data Structure Problem

Given:  $n$  cities  $(C_1 \dots C_n)$

|       |        |        |        |         |       |         |     |
|-------|--------|--------|--------|---------|-------|---------|-----|
| 1     | 2      | 3      | 4      | 5       | 6     | 7       | 8   |
| Delhi | Mumbai | B'lore | Jaipur | Kolkata | Dubai | Chennai | NYC |
|       |        | 400    |        | 700     |       |         |     |

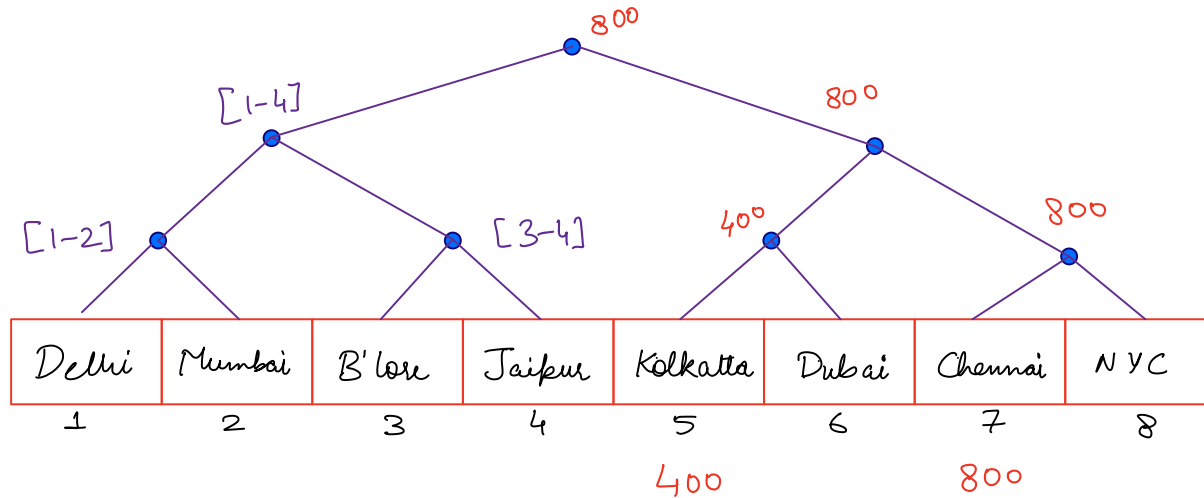
Dynamic:

New towers are being Built in arbitrary order.

Query:

What is height of tallest building in first  $i$  cities ?

# Data Structure Problem



In node  $[i-j]$  store = Height of tallest building in cities  $(c_i, c_{i+1}, \dots, c_j)$

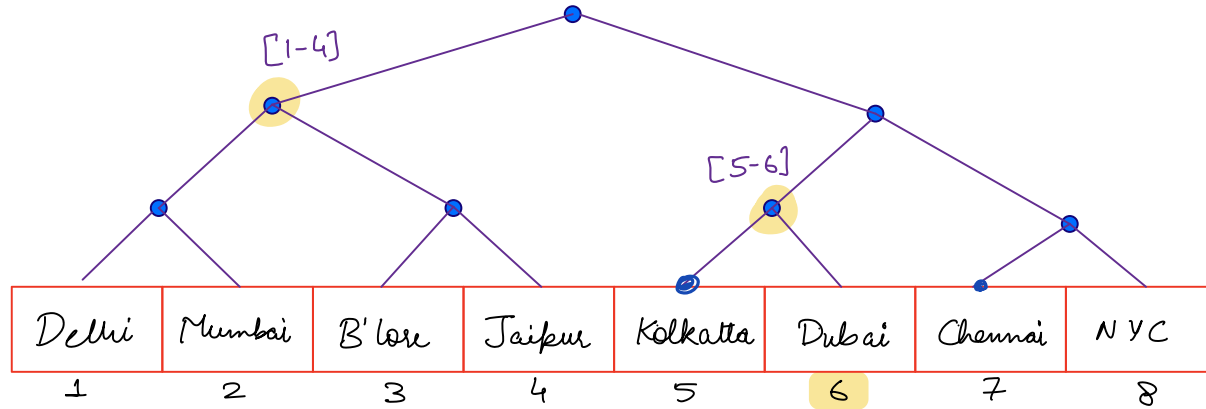
Ques 1:

How to update nodes of tree ?

$\log_2(n)$

# Data Structure Problem

$$6 = (110)_2$$



In node  $[i-j]$  store = Height of tallest building in cities  $(c_i, c_{i+1}, \dots, c_j)$

Ques 2:

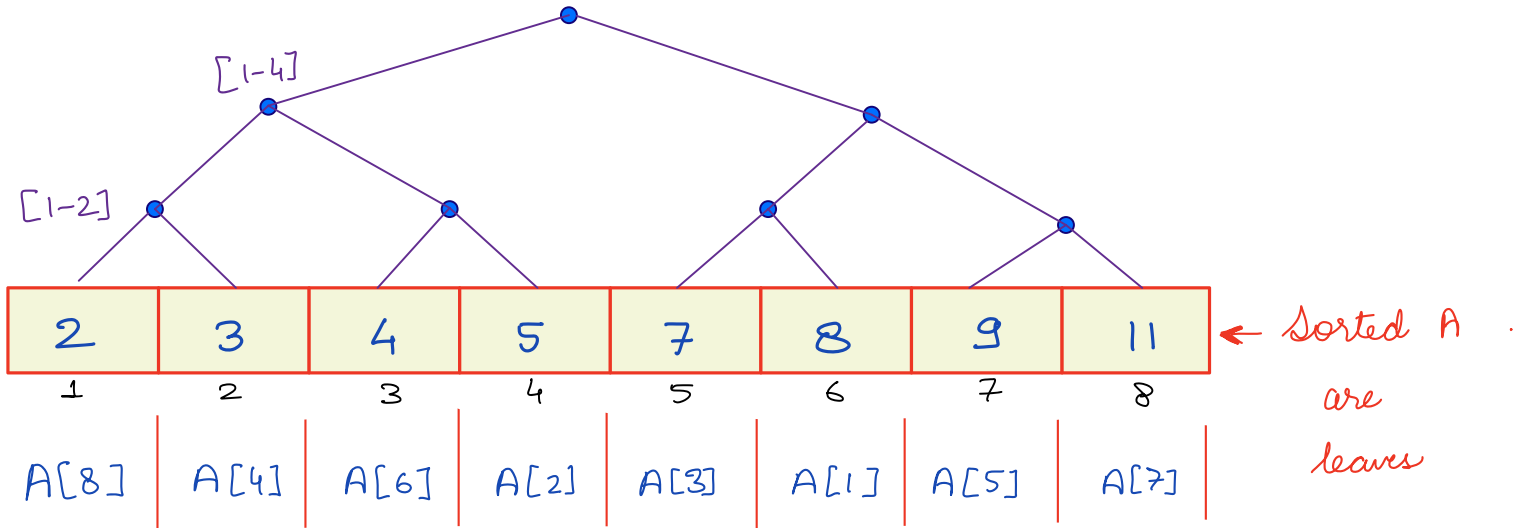
How to find height of tallest building in say first 6 cities?

$$\log_2(n)$$

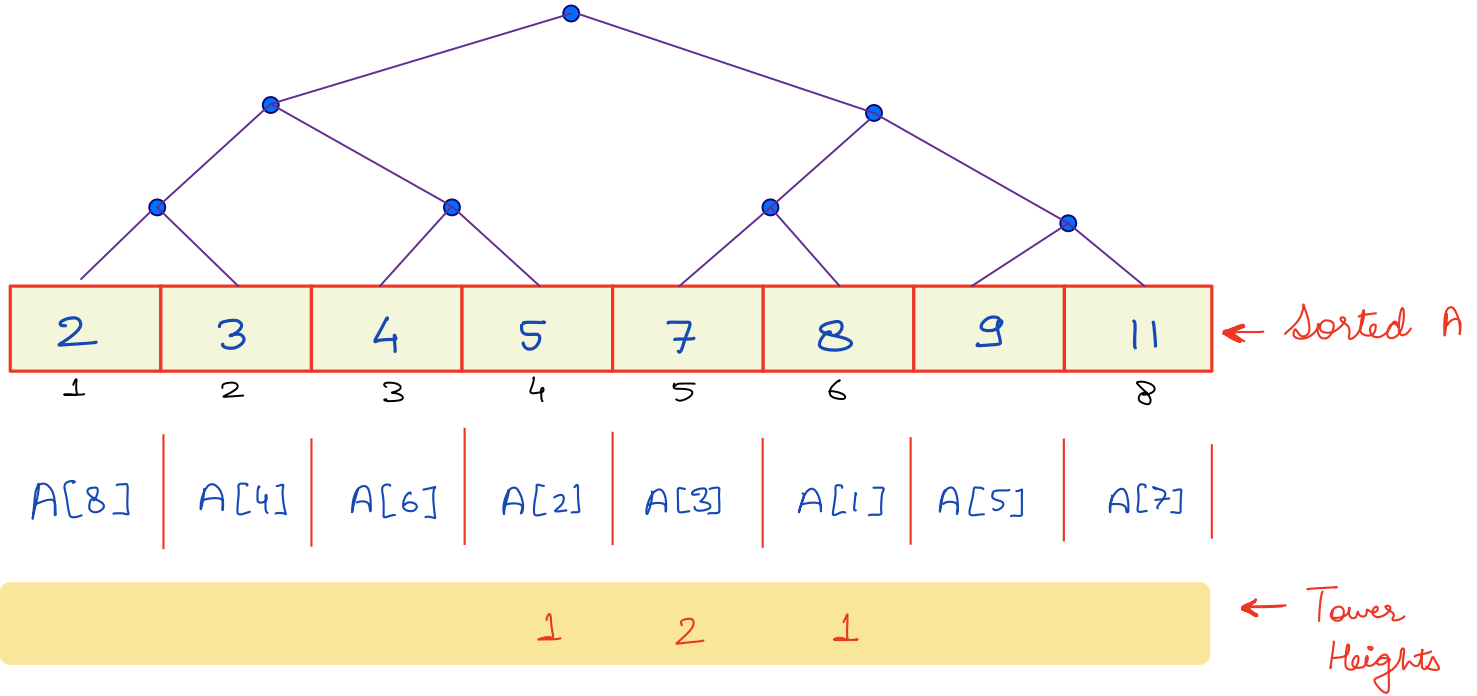


# Main Problem

$$L[i] = 1 + \max \left\{ L[j] \mid j < i, A[j] < A[i] \right\}$$



# Main Problem



$$A = \begin{bmatrix} 8 & 5 & 7 & 3 & 9 & 4 & 11 & 2 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 1 & 2 & 1 & 3 & 2 & 4 & 1 \end{bmatrix}$$

H.W. Prove that each  $L[i]$  is computable in  $O(\log n)$  time using our data-structure

# CHALLENGE PROBLEM

**Question:** What if elements in  $A$  are not **distinct**?