

# 2 D-Range Tree

M.B.Chandak

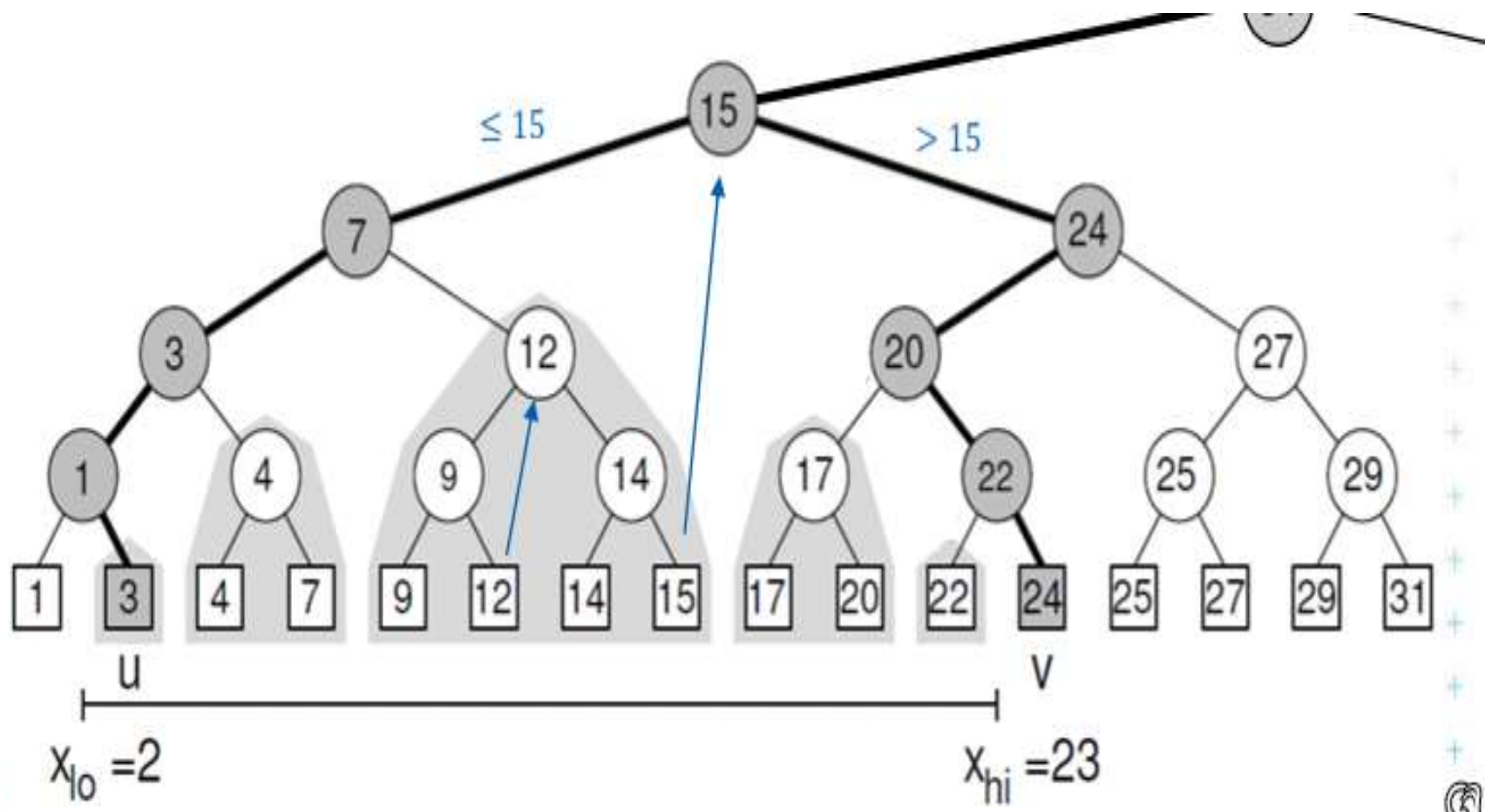
[hodcs@rknec.edu](mailto:hodcs@rknec.edu)

[www.mbchandak.com](http://www.mbchandak.com)

*Note: No voice or video sharing is permitted during Zoom meeting*

*All queries: please mail at [hodcs@rknec.edu](mailto:hodcs@rknec.edu)*

- **Unit-V**
- **Computational Geometry:** One Dimensional Range Searching, **Two Dimensional Range Searching**, Constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quad-trees, k-D Trees, Applications.



\*\*\*

৫

# What we have discussed in previous class

- One dimensional range tree.

Dataset = [10, 3, 5, 12, 16, 55, 23, 67 ....]

We can construct a tree in the form of BST, which can provide solution to queries like

Find values between: 12 and 50.

We have seen: some definitions also.

Part of sub tree is involved in search or some time complete sub tree is involved in search.

# Two dimensional Range Searching

Example data set

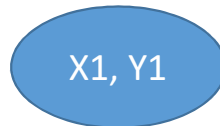
Name	Age	Salary

**Query: find age>x and salary>y**  
**Then 2D range search is suitable**  
**Data is in the form of <x,y> tuple.**  
**To optimise search, data should be properly stored.**

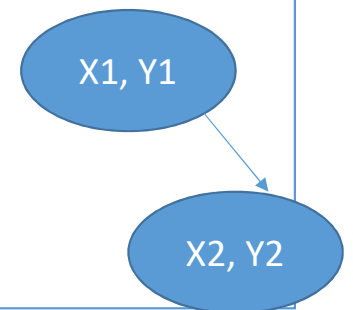
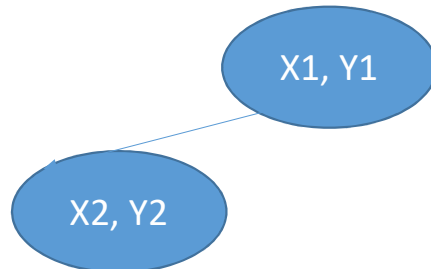
- There are two methods:
- 1. Unbalanced 2D range tree
- 2. Balanced 2D range tree.

## Example: Unbalanced 2D range tree

- $(x_1, y_1)$   $(x_2, y_2)$   $(x_3, y_3)$  .....  $(x_n, y_n)$
- Similar to BST → Select first value as Root



- For storing next value: Decision is X, Y, alternate compare operation.
- Compare X1 and X2 and store:  $(X_2, Y_2)$  → Either to left or right of  $(X_1, Y_1)$  based on  $X_1 > X_2$  or  $X_1 < X_2$

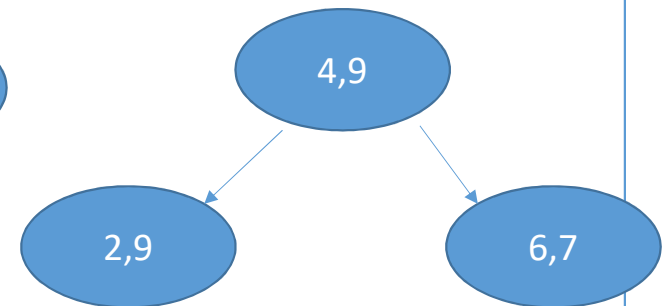
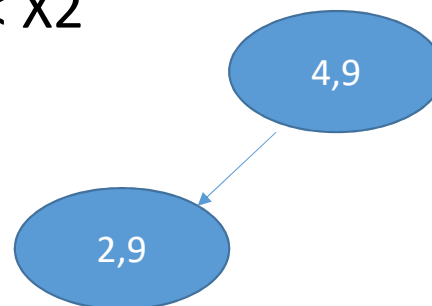
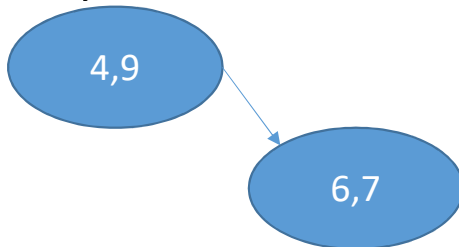


## Example: **Unbalanced** 2D range tree

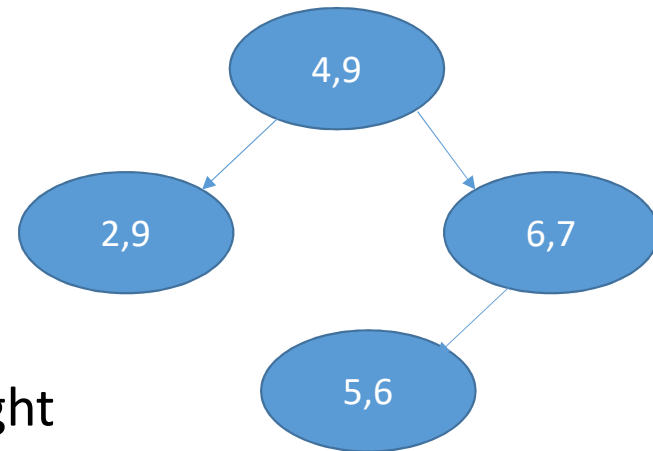
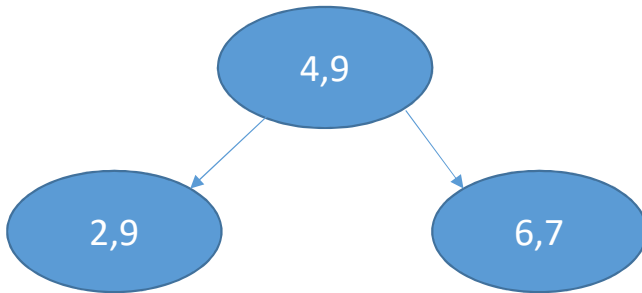
- **(4,9)** (6,7) (2,9) (5,6) (6,3) (8,5) (9,2) (10,2) (5,9) ..
- Similar to BST → **Select first value as Root**



- For storing next value: Decision is X, Y, alternate compare operation.
- Compare X1 and X2 and store: (X2, Y2) → Either to left or right of (X1, Y1) based on  $X1 > X2$  or  $X1 < X2$



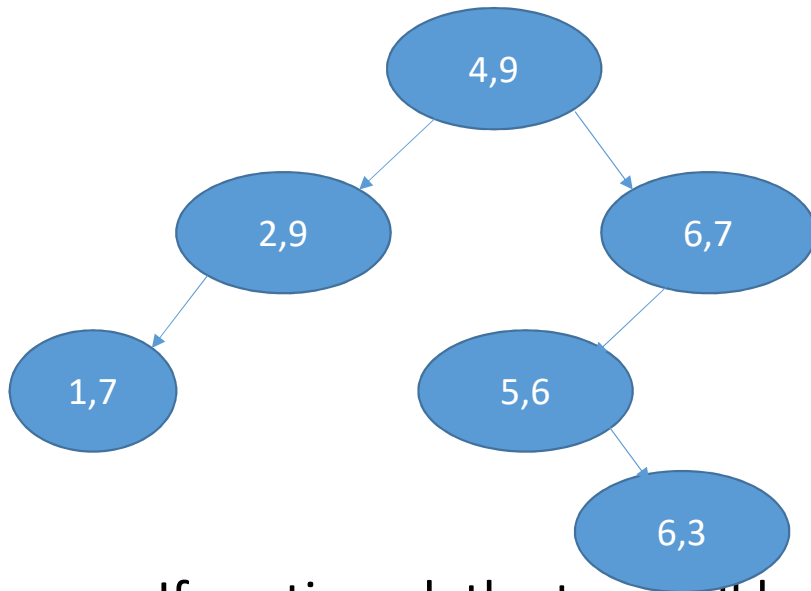
# Insert (5,6)



- First compare  $x_1=4$  with  $x_4=5$ ,  $\rightarrow$  Go Right
- Since location is not free: Compare  $y_2=7$  with  $y_4=6$   $\rightarrow$  Go left
- If (6,3) is to be inserted.  $\rightarrow$  (4:6) $\rightarrow$ Right
- Now compare: (7:3)  $\rightarrow$  Left
- Compare (5:6)  $\rightarrow$  Right [Insert as location is free]



Insert (6,3)    Insert new value (1,7)



- If continued: the tree will be unbalanced 2D Range tree.
- It may not guarantee the search time. The data positioning can be more precise, if tree is balanced. → Refer .xls sheet