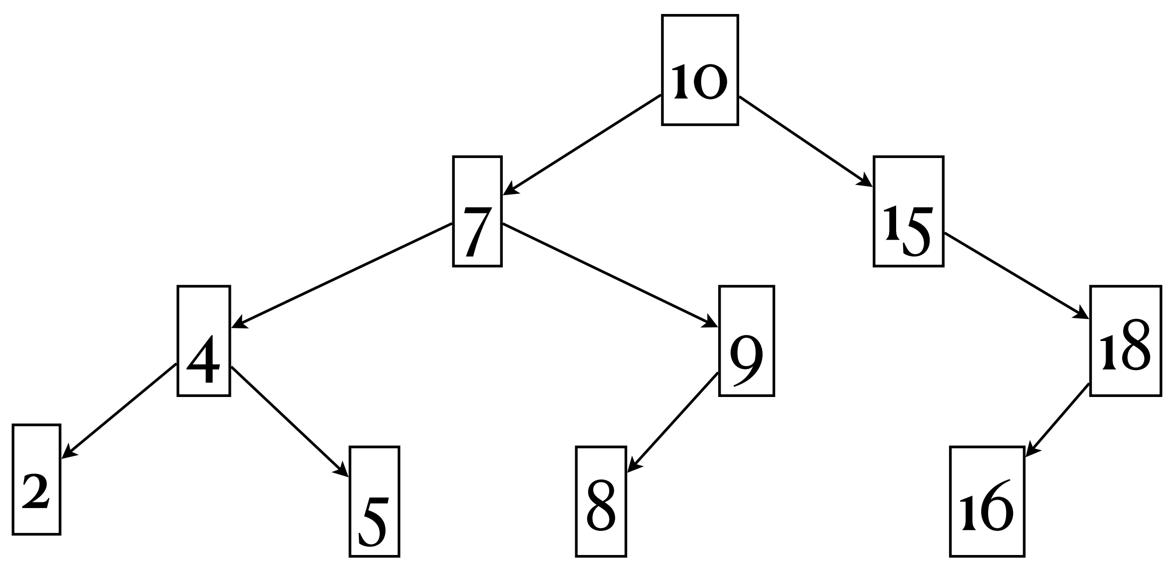
CSE 566 Spring 2023

Range Tree

Instructor: Mingfu Shao

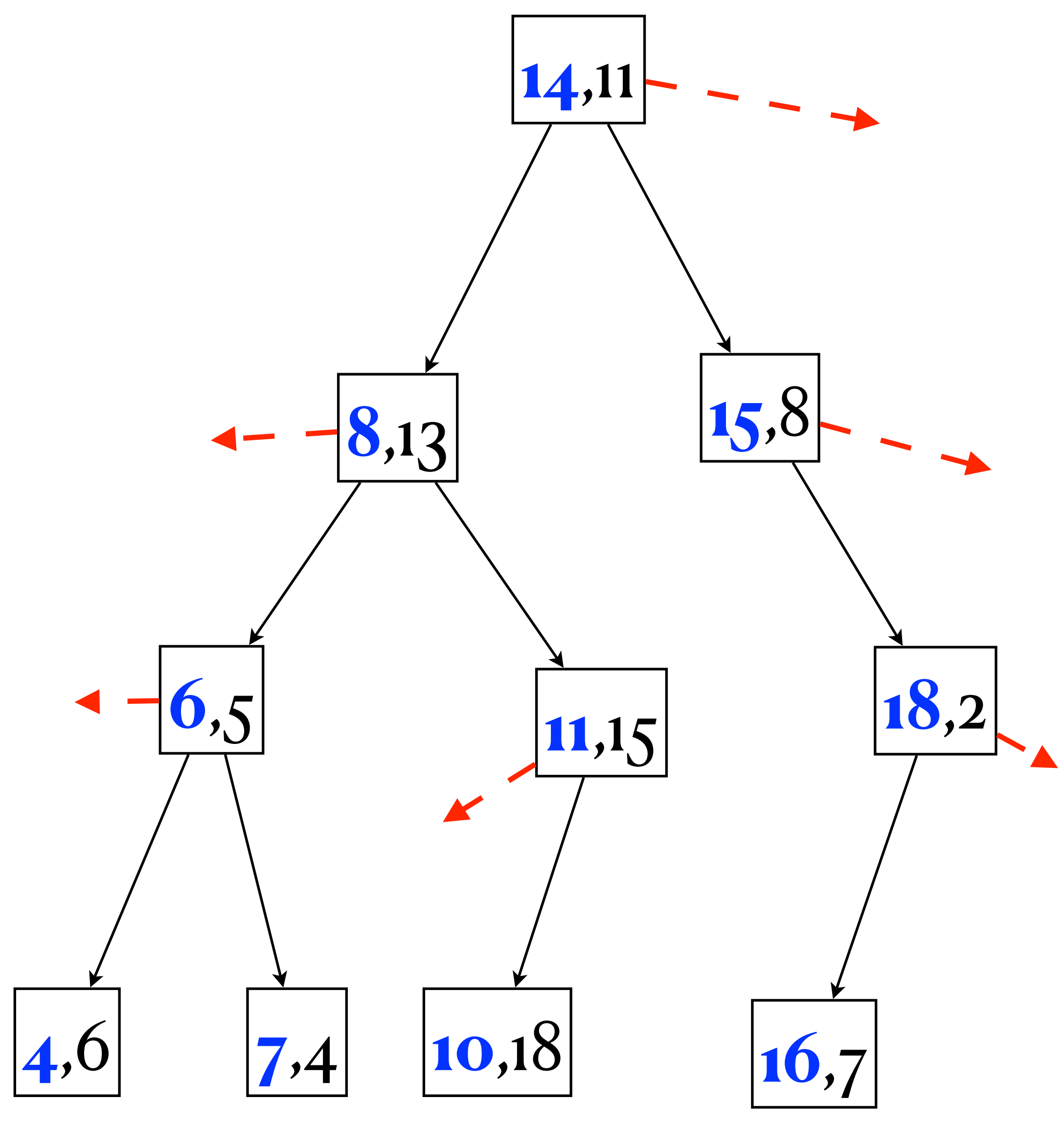
1D Range Tree

- Store a set of 1D items.
- 1D range tree is a balanced binary search tree.
- "Balanced" is defined as: the height is $O(\log n)$ with n nodes.

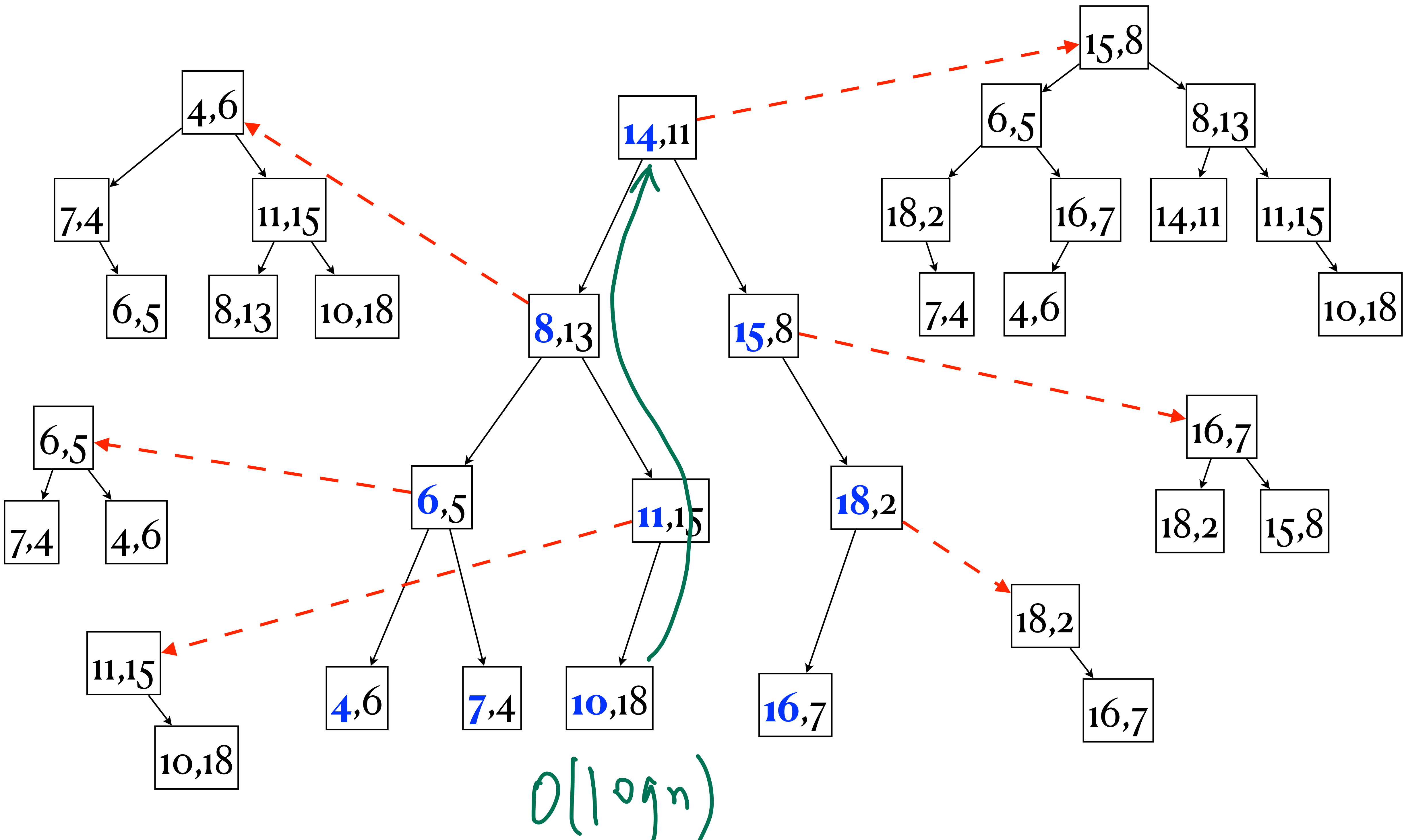


2D Range Tree

- Store a set 2D points P.
- The main tree, a balanced binary search tree T sorted by x-coordinates, includes all points in P.
- Each internal node v points to a balanced binary search tree sorted by y-coordinates that includes all points in the subtree rooted at v in T.



2D Range Tree: An Example



d-dimensional Range Tree

- Recursively defined, storing a set of points $P \in \mathbb{R}^d$
- Main tree T: a balanced binary search tree, sorted by the first-coordinates, includes in P.
- Each internal node v points to a (d-1)-dimensional range tree, sorted by the last (d-1) coordinates, that includes all points in the subtree rooted at v in T.

Space Complexity

- How many nodes in a 2D range tree, assuming $|P| = n$?

  - #times (an item appears) = O(height of T) = O(log n)

  #(total nodes) = $O(n \cdot \log n)$.

- How many nodes in a d-dimensional range tree, $|P| = n$?

#(total nodes) = $O\left(n \cdot \log^{d-1} n\right)$, by induction.

Constructing 2D Range Tree

- Construct range tree for a given set of 2D points *P*

```
PX <- sort P according to x
PY <- sort P according to y
function construct(PX, PY)
   create a node for median point m = PX[n/2]
   m.link <- build-binary-search-tree(PY)
   partition PX into (PX1, m, PX2), where PX1.x < m.x < PX2.x
   partition PY into (PY1, m, PY2), where PY1.x < m.x < PY2.x
   m.left-child <- construct(PX1, PY1)
   m.right-child <- construct(PX2, PY2)
   return node
end function
```

$\Theta(n \cdot \log n)$

$\Theta(n)$

$T(n) = \Theta(n) + 2T(n/2)$

$\Rightarrow T(n) = \Theta(n \cdot \log n)$

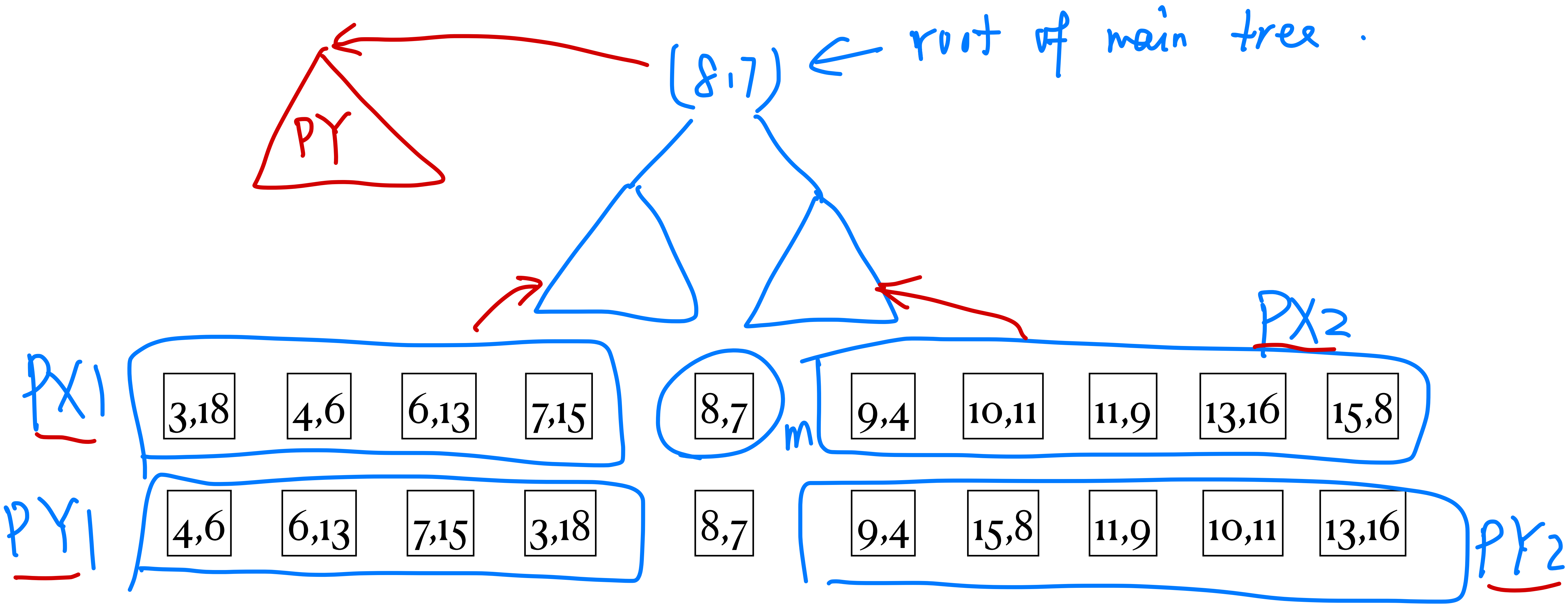
running time:

$\Theta(n \cdot \log n)$.

An Example

PX: | 3,18 | 4,6 | 6,13 | 7,15 | 8,7 | 9,4 | 10,11 | 11,9 | 13,16 | 15,8 |

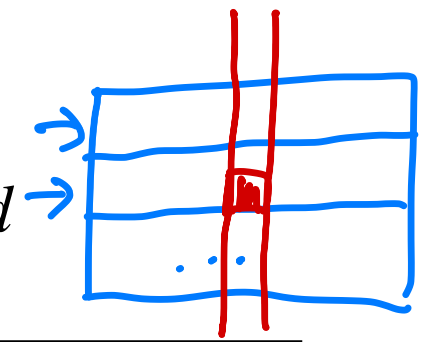
PY: | 9,4 | 4,6 | 8,7 | 15,8 | 11,9 | 10,11 | 6,13 | 7,15 | 13,16 | 3,18 |



Constructing d-dim Range Tree

- Construct range tree for a given set of points $P \in \mathscr{R}^d$

PD



```
sort P according to each dimension -> PD, matrix of size d by n
function construct(PD, cd)
  m = PD[cd, n/2]
  create a new node for point m
  if(cd < d): m.link <- construct(PD, cd+1)
  partition PD into (PD1, m, PD2), where PD1.cd < m.cd < PD2.cd
  m.left-child <- construct(PD1, cd)
  m.right-child <- construct(PD2, cd)
  return node
end function
```

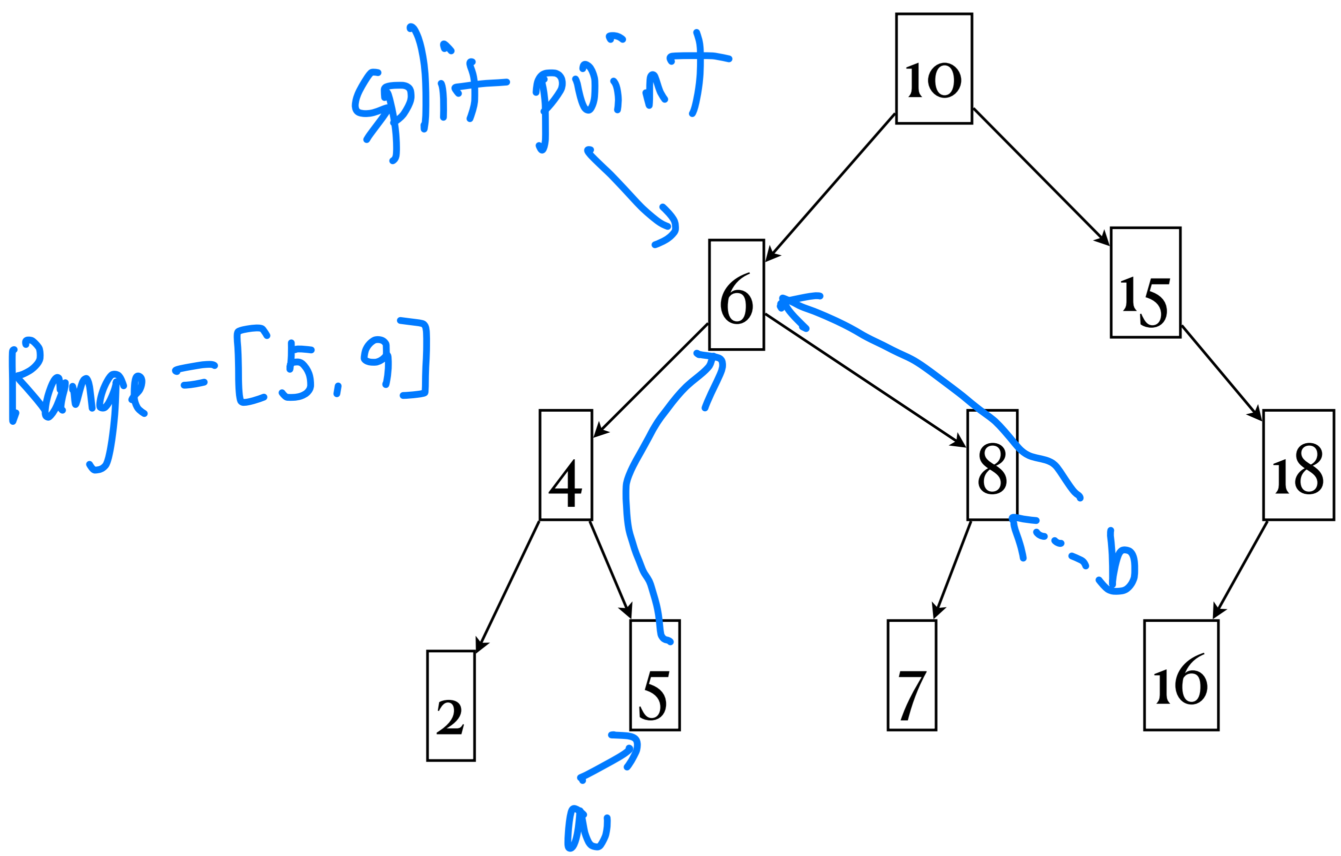
$d \cdot n \cdot \log n$

$T(n,k) = T(n,k-1) + nd + 2T(n/2, k)$

$\Rightarrow \quad T(n,k) = O[n \cdot \log^{d-1} n)$

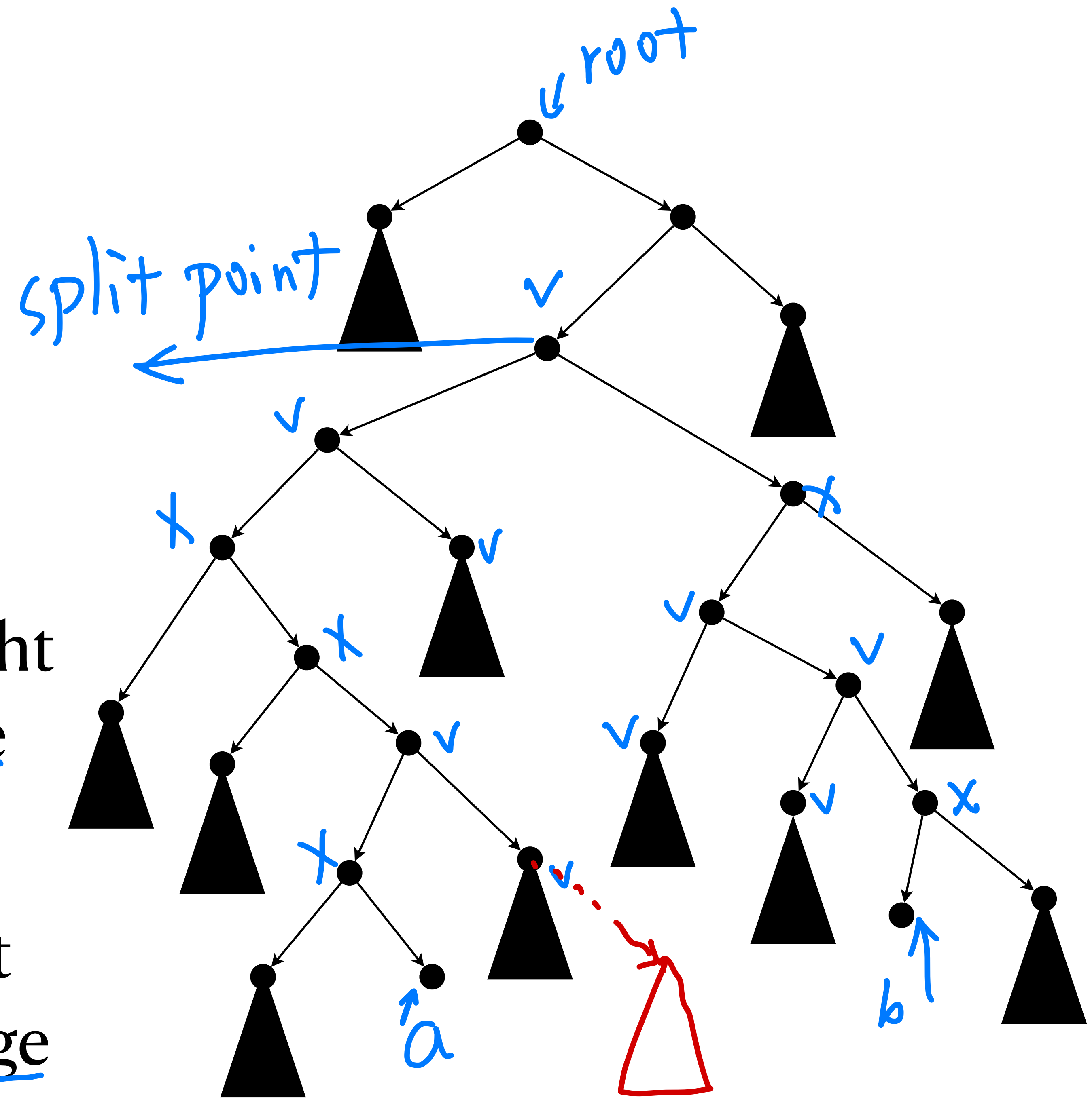
1D Range Query

- Find points in range $[a, b]$ in a balanced binary search tree.
- Running time: $O(\log n + m)$

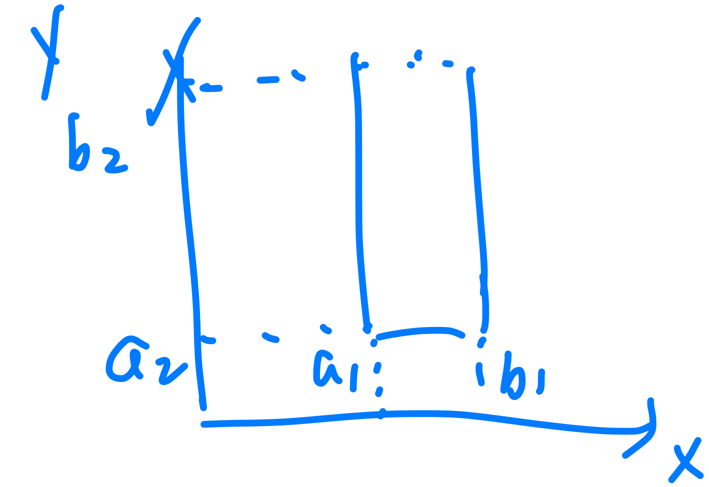


1D Range Query

- Find the split point v
- In the left subtree find a
- In the right subtree find b
- Follow the path from a to v, include the node and the right subtree if using the left-edge
- Follow the path from b to v, include the node and the left subtree if using the right-edge



2D Range Query



- Find points in the range $[a_1, b_1]$ and $[a_2, b_2]$ in a range tree, i.e., points (x, y) with $a_1 \leq x \leq b_1$ and $a_2 \leq y \leq b_2$.

    1. Do 1D range query $[a_1, b_1]$ on the main tree T, which returns a set of nodes and a set of subtrees.

$O(\log n)$

    2. Verify individual nodes.
    3. Recursively do 1D range query $[a_2, b_2]$ on each individual subtree. (pointed by the root of the subtrees)

- Running time:

$O(\log^2 n + m)$

d-dimensional Range Query

- Given hyper-rectangle specified by $[a_i, b_i]$, find all points $(x_1, x_2, \cdots, x_d)$ satisfying $a_i \leq x_i \leq b_i$ for all $1 \leq i \leq d$.
    1. Do 1D range query $[a_1, b_1]$ on the main tree T, which returns a set of nodes and a set of (d-1)-range trees.
    2. Verify individual nodes.
    3. Recursively do range query on $[a_i, b_i]$, $2 \leq i \leq d$, for each (d-1)-range tree.
- Running time: $O(\log^d n + m)$.

kd-tree vs. Range Tree

	kd-tree (2D)	range tree (2D)
space complexity	$O(n)$	$\boxed{O(n \log n)}$
construction	$O(n \log n)$	$O(n \log n)$
insert a point	$O(\log n)$	$O(\log^2 n)$
range query	$O(\sqrt{n} + m)$	$O(\log^2 n + m)$