CSE 100:
KD TREES

Thanks to Dylan McNamara, Wikipedia, and the textbook “Data Structures and Algorithms in C++”

Announcements

- Iclicker Reminder
- Break from BST's today for the project. We'll return to BSTs next class

- Reading Quiz due 1 hour before the class M/W/F (3pm)
- Midterm Exams: Wed (10/24) and Monday (11/19)
- Office Hours (iClicker choice):
  - A – M/W – 10-11am
  - B – T/Th – 4-5pm
- PA1
  - Checkpoint Deadline: 11:59pm on Thursday, 10/11 (not eligible for slip day)
  - Final Deadline: 11:59pm on Thursday, 10/18 (slip day eligible)

Goals for today

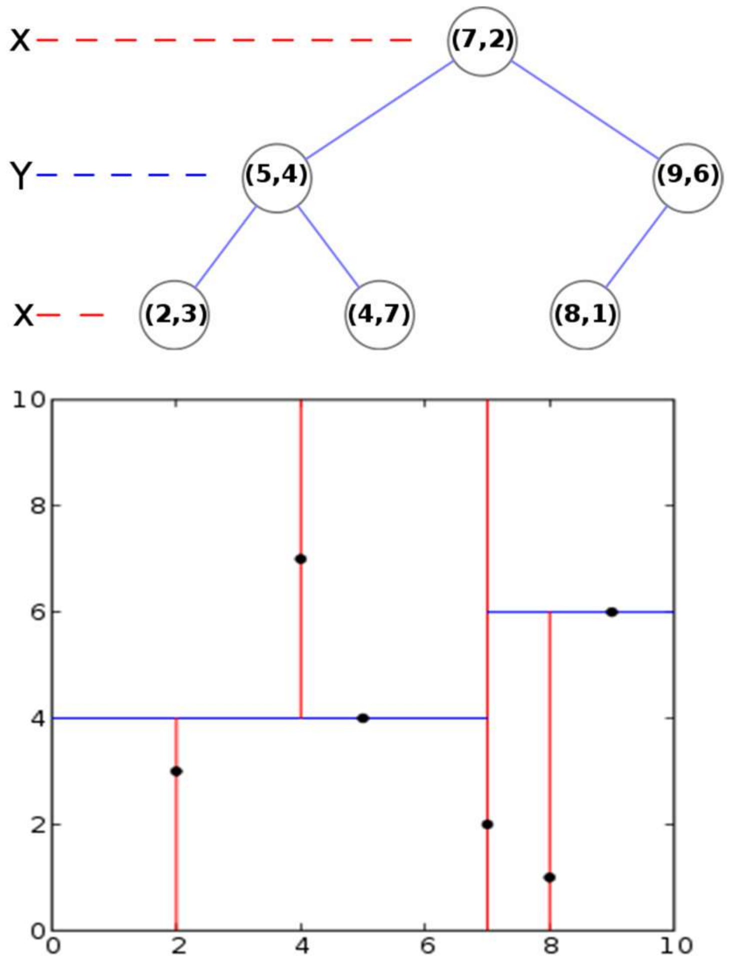
- Recognize the value of KD Trees
- Implement methods in a KD Tree:
  - Insert
  - Find
  - Build
  - FindRange
  - NearestNeighbor

KD Tree

- How would you implement < on a multi-dimensional Point?

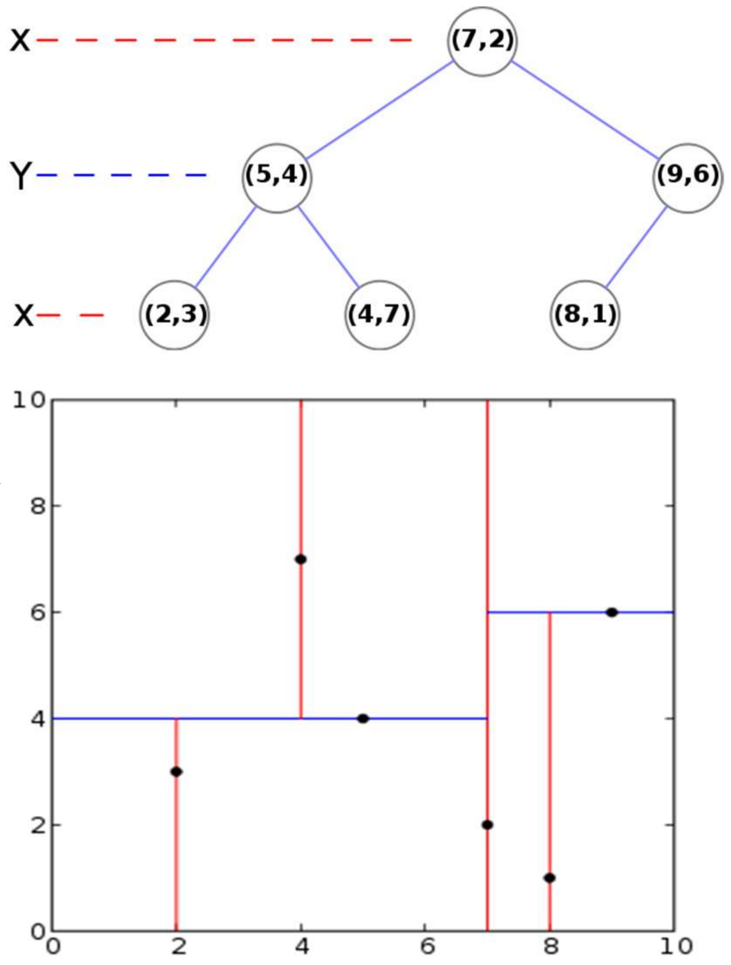
KD Tree

- How would you implement < on a multi-dimensional Point?
  - You can't… Hence, KD-Trees
- Each level is a different dimension (e.g. x, y, x, y,…)
  - left.dim < root.dim < right.dim

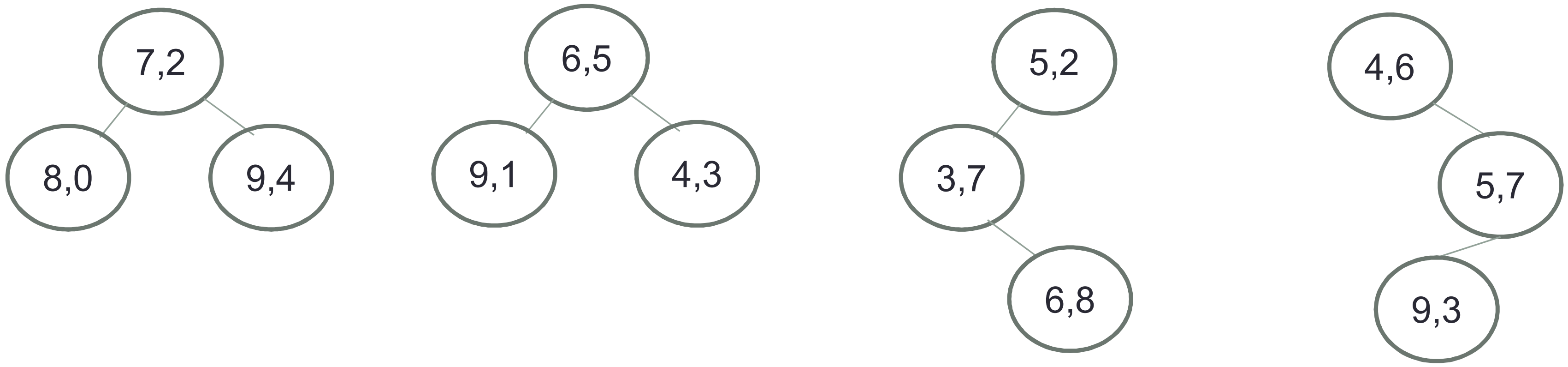


KD Tree

- How would you implement < on a multi-dimensional Point?
  - You can't… Hence, KD-Trees
- Each level is a different dimension (e.g. x, y, x, y,…)
  - left.dim < root.dim < right.dim
- Why?
  - Good for find in range, find nearest neighbor
  - Graphics, Vision, ML



KD Tree

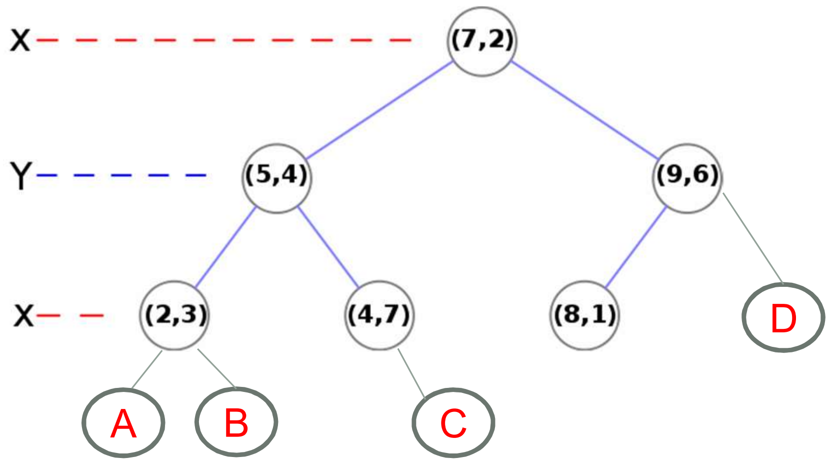


- How many of these trees are valid KD Trees?

A. 1
B. 2
[✓ C. 3]
D. 4

KD Tree Insert

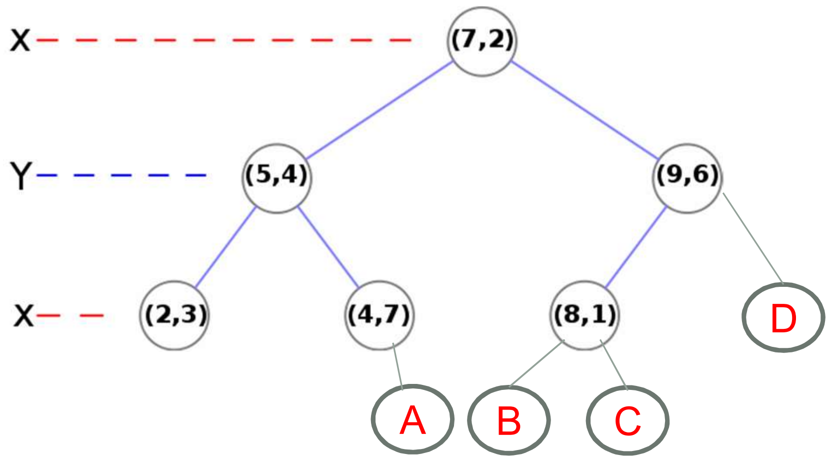
- Suppose you wish to insert the point (6,3), where would it go?



E – None of the above, need to rebalance

KD Tree Insert

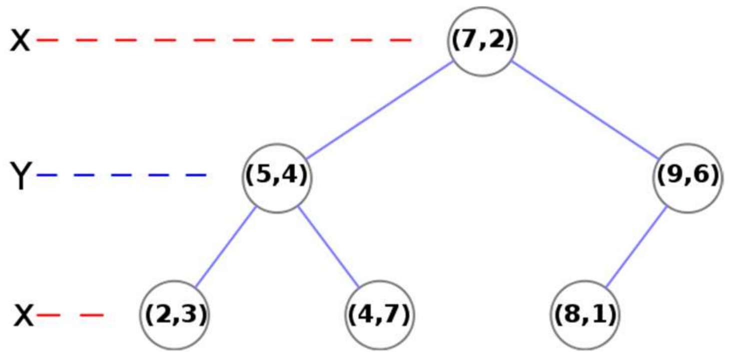
- Suppose you wish to insert the point (7,8), where would it go?



E – Either A or D

KD Tree Build

- No guarantee the tree will be balanced. But if you know all the points a-priori, you can build a fairly balanced tree*



*duplicate x or y values can cause some imbalance depending on implementation

KD Tree Build

Input: list of items, start, end, dimension

Output: root of subtree

BuildRecurse:

```
    Sort items from start to (end-1) over dimension
    Select (leftmost) median for index mid
    Create Node p holding median
    p->left = BuildRecurse(list,start,mid,toggle_dim)
    p->right = BuildRecurse(list,mid+1,end, toggle_dim)
    return p
```

KD Tree Build

Input: list of items, start, end, dimension

Output: root of subtree

BuildRecurse:

```
    Sort items from start to (end-1) over dimension
    Select (leftmost) median for index mid
    Create Node p holding median
    p->left = BuildRecurse(list,start,mid,toggle_dim)
    p->right = BuildRecurse(list,mid+1,end, toggle_dim)
    return p
```

Separate paper, build a tree with these points with your group:
(1.0, 3.2), (3.2, 1.0), (5.7, 3.2), (1.8, 2.9), (4.4, 4.2), (0.0, 0.0), (2.7, 9.1)
-- start with x dimension

KD Tree Build

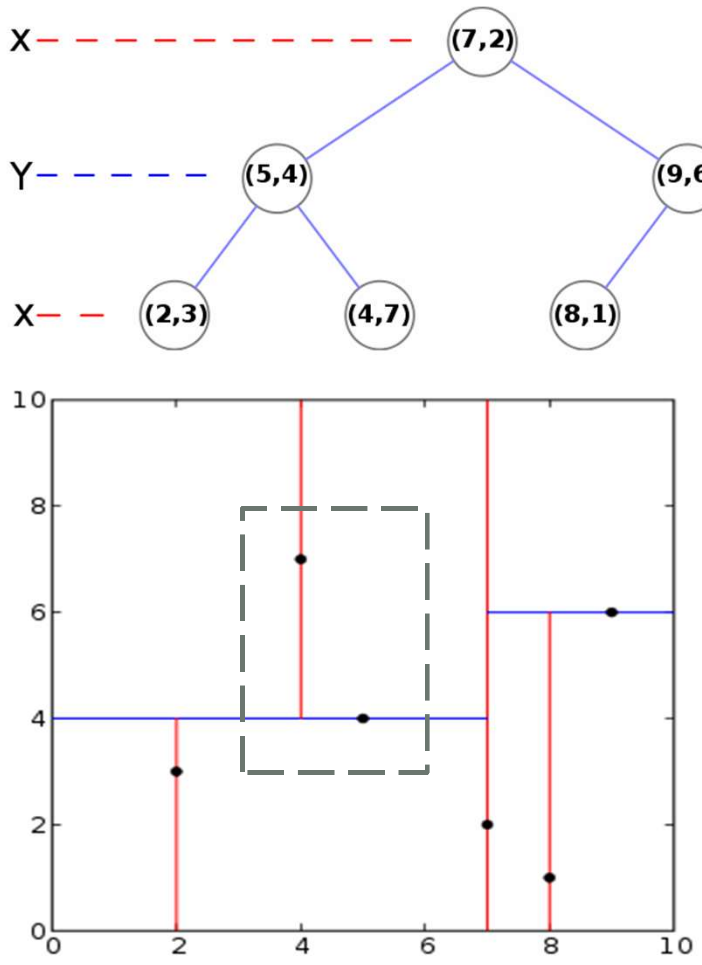
(1.0, 3.2), (3.2, 1.0), (5.7, 3.2), (1.8, 2.9), (4.4, 4.2), (0.0, 0.0), (2.7, 9.1)

KD Tree Range Search

- Intuition:

-Only explore a path of the tree if it could be in the range

-Add nodes as they appear in the path



KD Tree Range Search

- Intuition:

-Only explore a path of the tree if it could be in the range

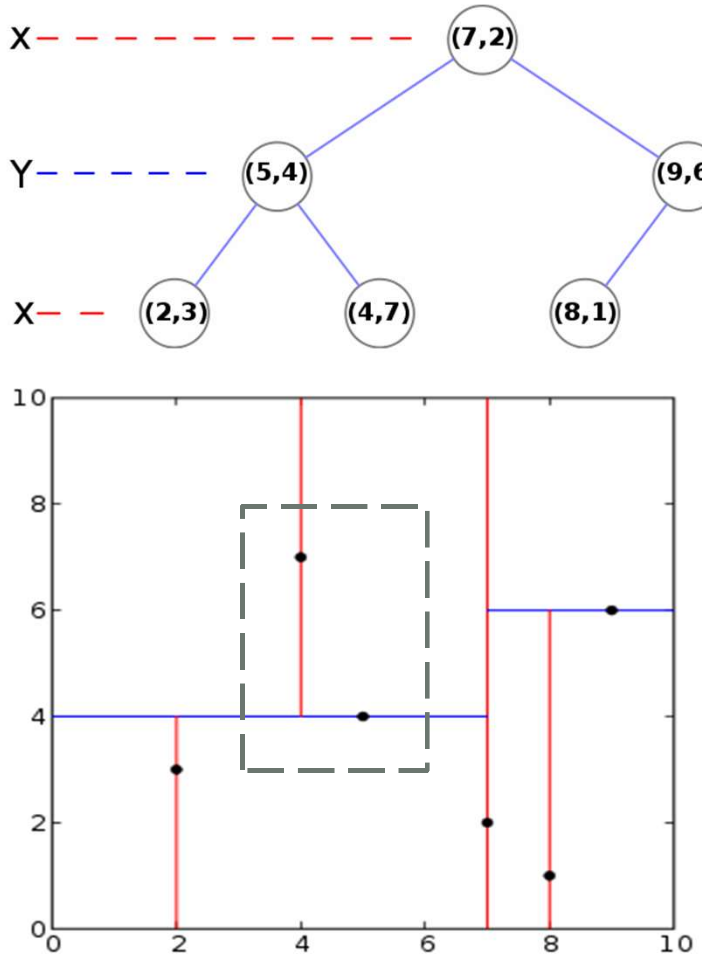
-Add nodes as they appear in the path

In looking for this range: (3<=x<=6, 3<=y<=8), would you explore the right subtree of (7,2)?

A. Yes

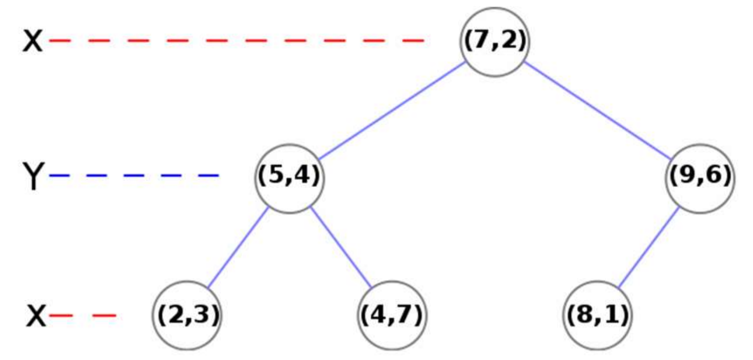
B. No

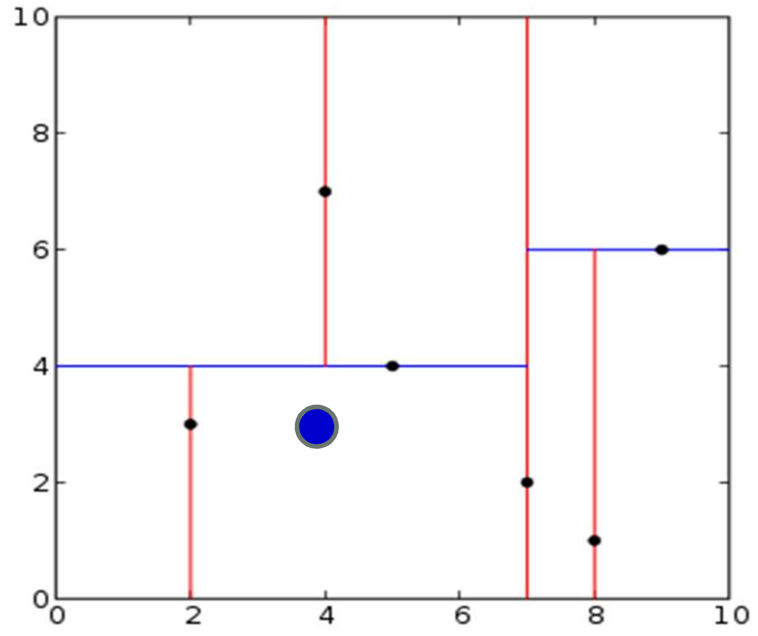
[✓ C. Maybe]

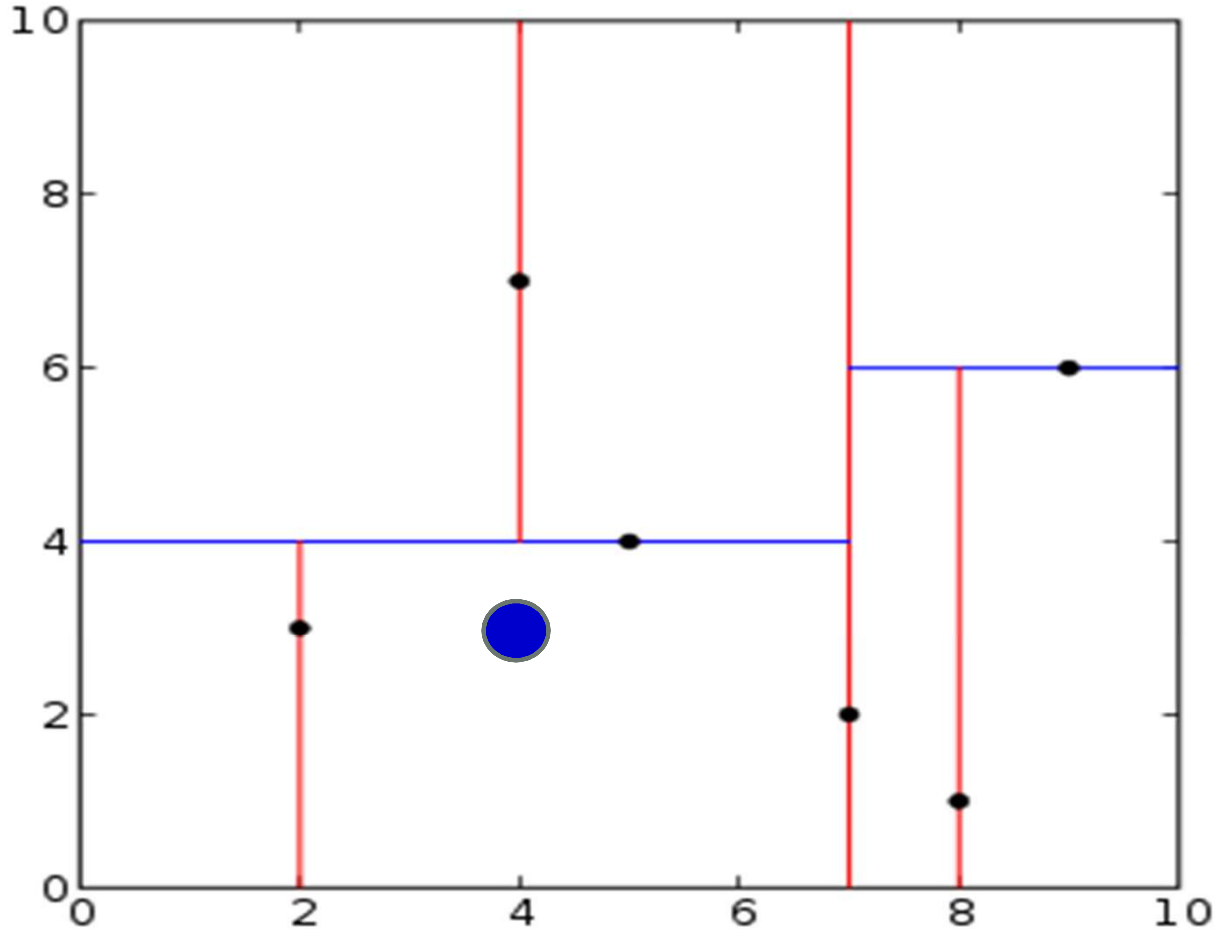


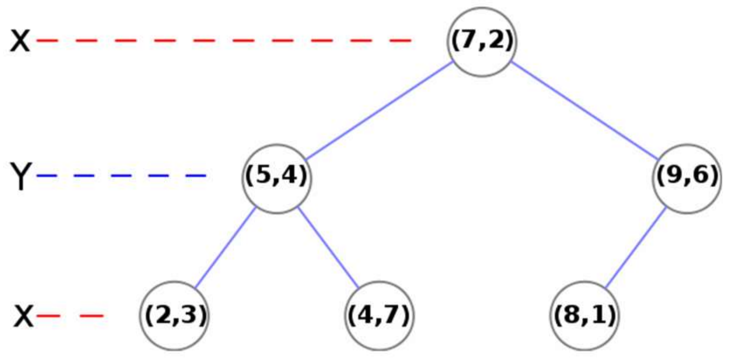
KD Tree Nearest Neighbor

- Intuition:
  - Make your best guess similar to find.
  - Only explore alternative paths if they might produce a better result



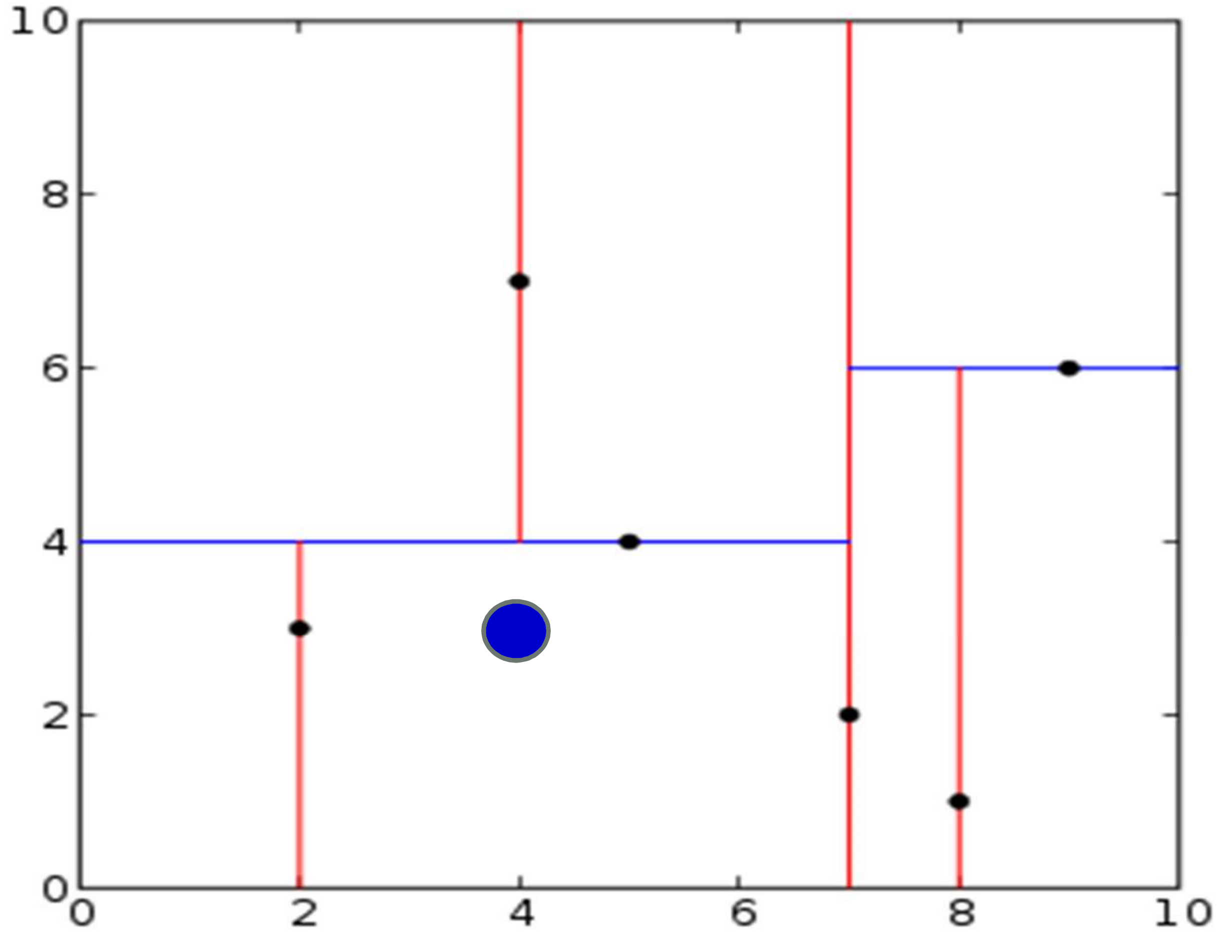


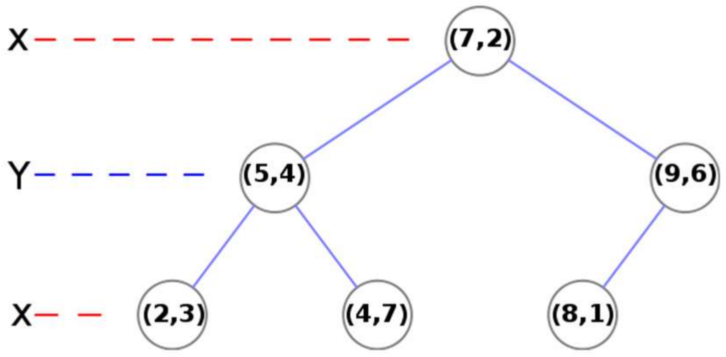


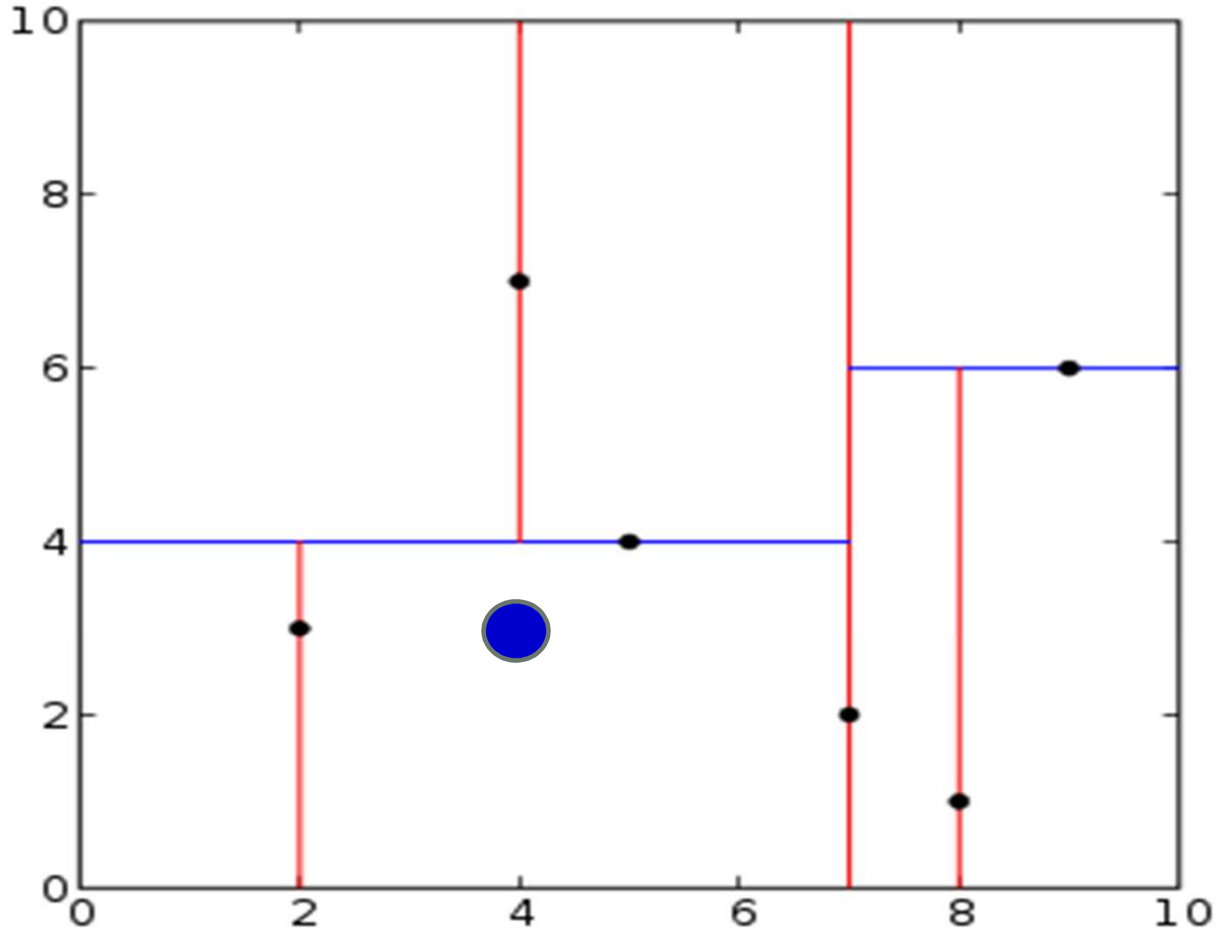
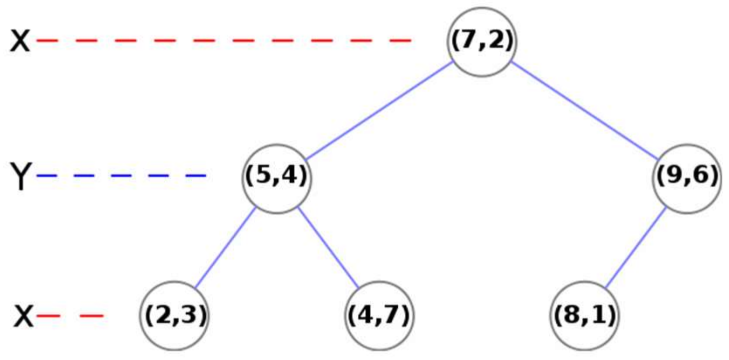


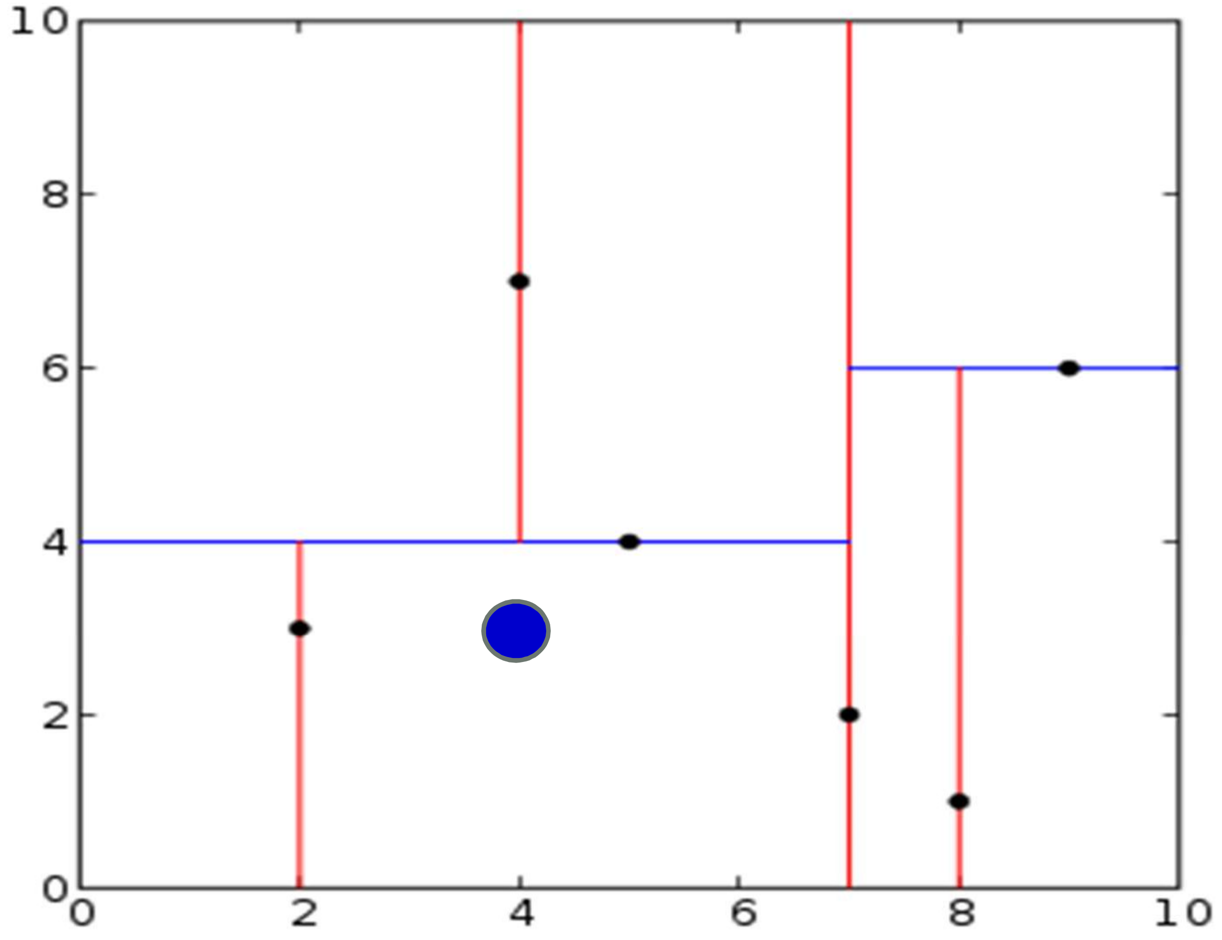
- Where would the algorithm stop if we run **find** on the point (4,3)?

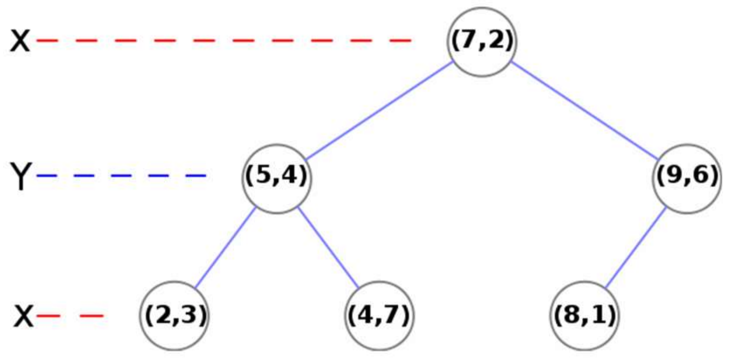
A. (2,3)
B. (4,7)
C. (5,4)
[✓ D. (7,2)]
E. None of the above


0
2
4
6
8
10


X
Y
X
(7,2)
(5,4)
(9,6)
(2,3)
(4,7)
(8,1)



X
(7,2)
Y
(5,4)
(9,6)
X
(2,3)
(4,7)
(8,1)

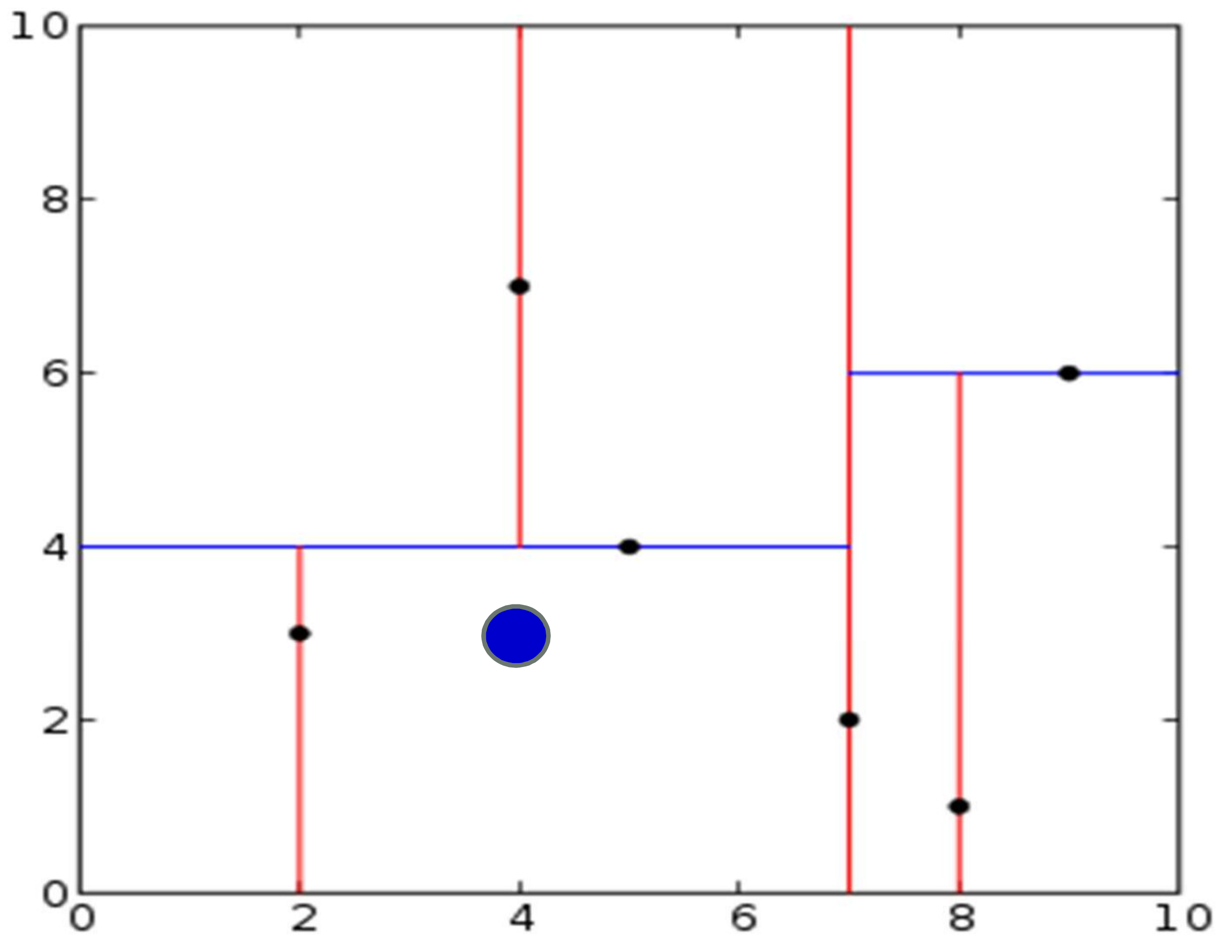


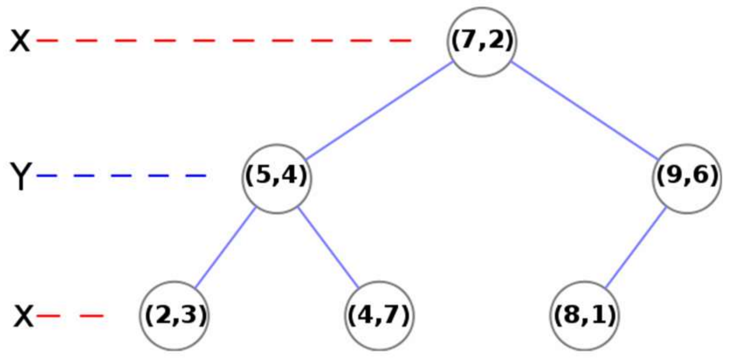


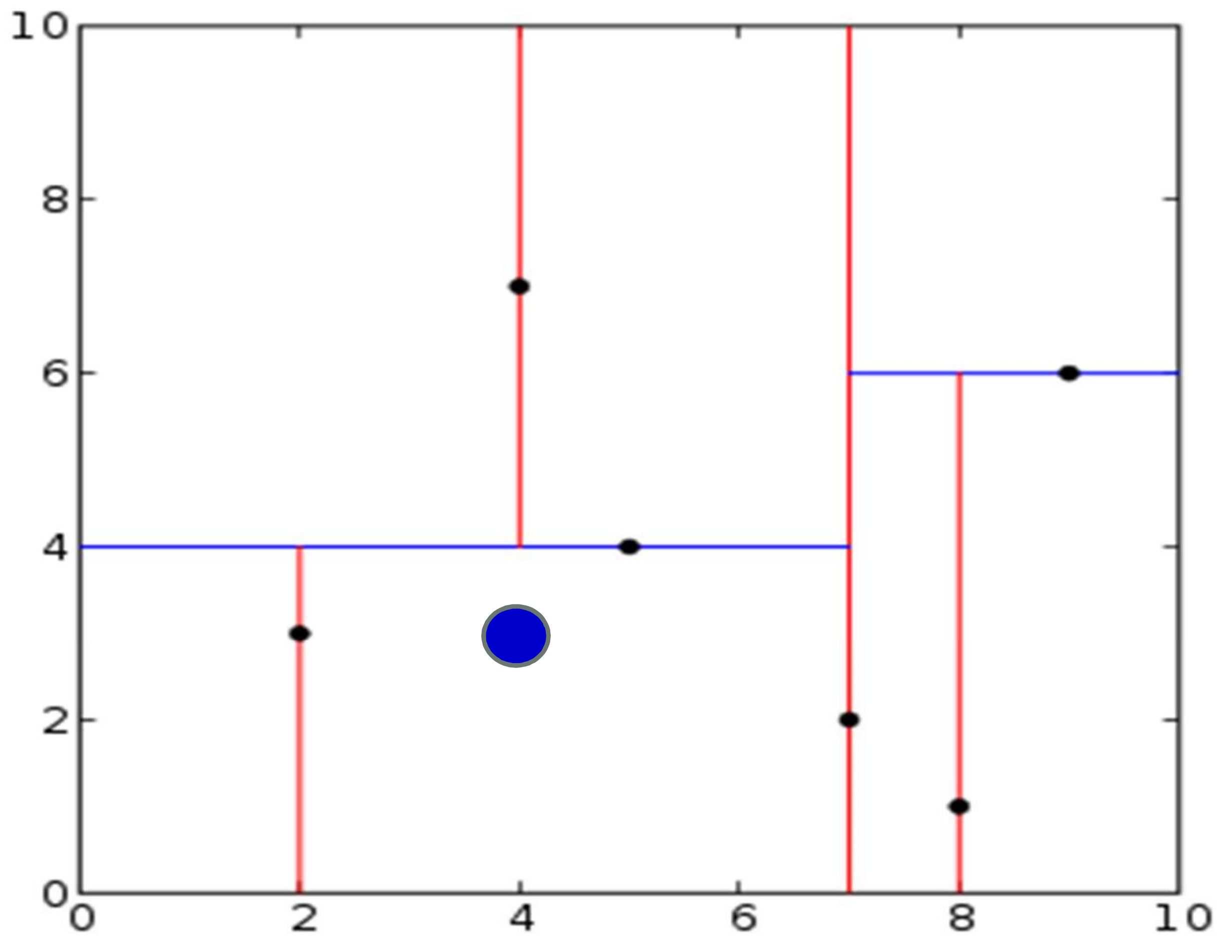
Do we need to check the right subtree of (5,4)?

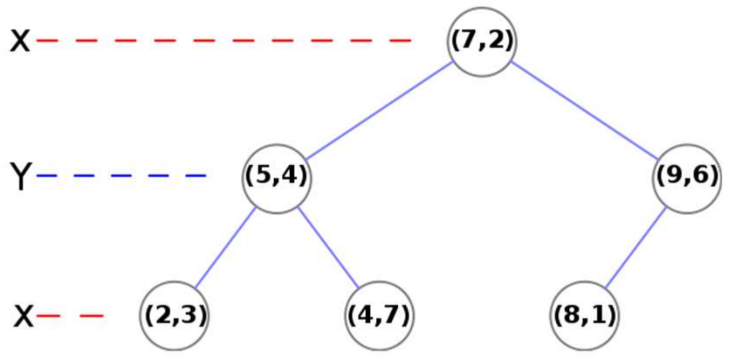
A. Yes

B. No


10
8
6
4
2
0
0
2
4
6
8
10


X
Y
X
(7,2)
(5,4)
(9,6)
(2,3)
(4,7)
(8,1)





Do we need to check the right subtree of (7,2)?

[✓ A. Yes]

B. No

PA1 Part 2

- Implement:

Build

Find nearest neighbor

Main2 (mainly dealing with i/o)