```
Show the KD-Tree that results after deleting the point (35,60) from the
original tree. Show both the tree and substructure.

|files|
```

↳Add **Question 5.3**

✚ Add **Question 6**

Save

Q1 Let's start slowly

10 Points

Some True/False for you!

Q1.1 Binary Search Anyone?

5 Points

If a node in a 2-dimensional KD-Tree has $x = 20$, all of the nodes in its left subtree must have $x < 20$.

- ○ True
- ◉ False

Q1.2 How can I abstract this?

5 Points

A perfectly balanced 2-Dimensional KD-tree can be built in $O(n \log n)$ time.

- ◉ True
- ○ False

Q2 This one will be tricky part 2

25 Points

Provide a proof of the following Theorem:
Given a balanced KD-tree with n points, range counting queries can be answered in $O(\sqrt{n})$ time.

Here is Lemma for you to use in your proof: Given a balanced KD-tree with n points, any vertical or horizontal line stabs $O(\sqrt{n})$ cells of the tree.

No files uploaded

Q3 Time to explain your thinking.

15 Points

What is the worst-case scenario for searching in a KD-Tree?
This does not mean give an analysis of run time.
Think about how the points would have to be structured around the query point to make it as long as possible to find the nearest neighbor.

Q4 KD tree Insertion with 4 dimensional points

20 Points

Add the following 10 nodes to an **initially empty KD tree**.

(20, 7, 8, -1), (10, 50, 28, 70), (15, 51, 28, 70),
(30, 0, 2, 3), (25, 1, 2, 3), (15, 51, 28, 60),
(15, 51, 28, 50), (16, 51, 30, 0), (10, 60, 30, 50),
(3, 55, 40, 58)

Q4.1 What is the left child of Node (20,7,8,-1)?

4 Points

- ○ (20, 7, 8, -1)
- ◉ (10, 50, 28, 70)
- ○ (15, 51, 28, 70)
- ○ (30, 0, 2, 3)
- ○ (25, 1, 2, 3)
- ○ (15, 51, 28, 60)
- ○ (15, 51, 28, 50)
- ○ (16, 51, 30, 0)
- ○ (10, 60, 30, 50)
- ○ (3, 55, 40, 58)

Q4.2 What is the right child of Node (20,7,8,-1)?

4 Points

- ○ (20, 7, 8, -1)
- ○ (10, 50, 28, 70)
- ○ (15, 51, 28, 70)
- ◉ (30, 0, 2, 3)
- ○ (25, 1, 2, 3)
- ○ (15, 51, 28, 60)
- ○ (15, 51, 28, 50)
- ○ (16, 51, 30, 0)
- ○ (10, 60, 30, 50)
- ○ (3, 55, 40, 58)

Q4.3 What is the left child of Node (15,51,28,50)?

4 Points

- ○ (20, 7, 8, -1)
- ○ (10, 50, 28, 70)
- ○ (15, 51, 28, 70)
- ○ (30, 0, 2, 3)
- ○ (25, 1, 2, 3)
- ○ (15, 51, 28, 60)
- ○ (15, 51, 28, 50)
- ○ (16, 51, 30, 0)
- ◉ (10, 60, 30, 50)
- ○ (3, 55, 40, 58)

Q4.4 What is the right child of Node (15,51,28,50)?

4 Points

- ○ (20, 7, 8, -1)
- ○ (10, 50, 28, 70)
- ○ (15, 51, 28, 70)
- ○ (30, 0, 2, 3)
- ○ (25, 1, 2, 3)
- ○ (15, 51, 28, 60)
- ○ (15, 51, 28, 50)
- ◉ (16, 51, 30, 0)
- ○ (10, 60, 30, 50)
- ○ (3, 55, 40, 58)

Q4.5 What is the left child of Node (15,51,28,60)?

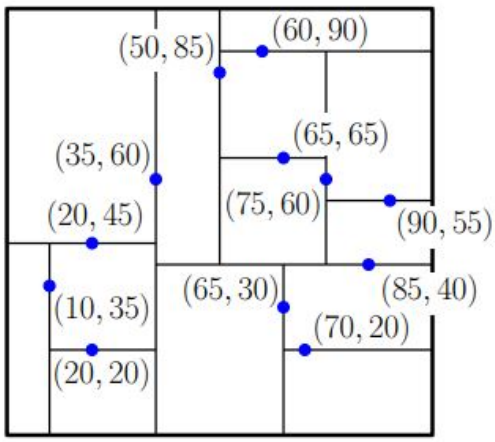
4 Points

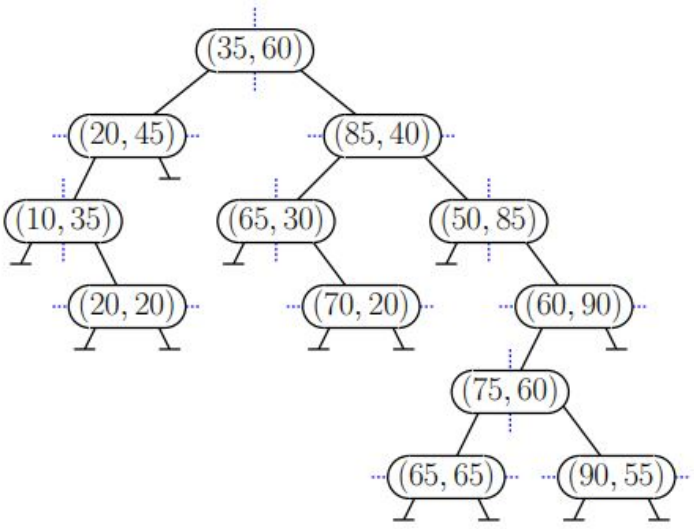
- ○ (20, 7, 8, -1)
- ○ (10, 50, 28, 70)
- ○ (15, 51, 28, 70)
- ○ (30, 0, 2, 3)
- ○ (25, 1, 2, 3)
- ○ (15, 51, 28, 60)
- ◉ (15, 51, 28, 50)
- ○ (16, 51, 30, 0)
- ○ (10, 60, 30, 50)
- ○ (3, 55, 40, 58)

Q5 Insertion and Deletion

30 Points

Use the following tree for both of the problems:





Q5.1 First Insert

15 Points

Show the result of inserting (80,10) into this tree. Show both the tree and substructure.

No files uploaded

Q5.2 Then Delete

15 Points

Show the KD-Tree that results after deleting the point (35,60) from the original tree. Show both the tree and substructure.

No files uploaded