CS210 Discussion

Week 10

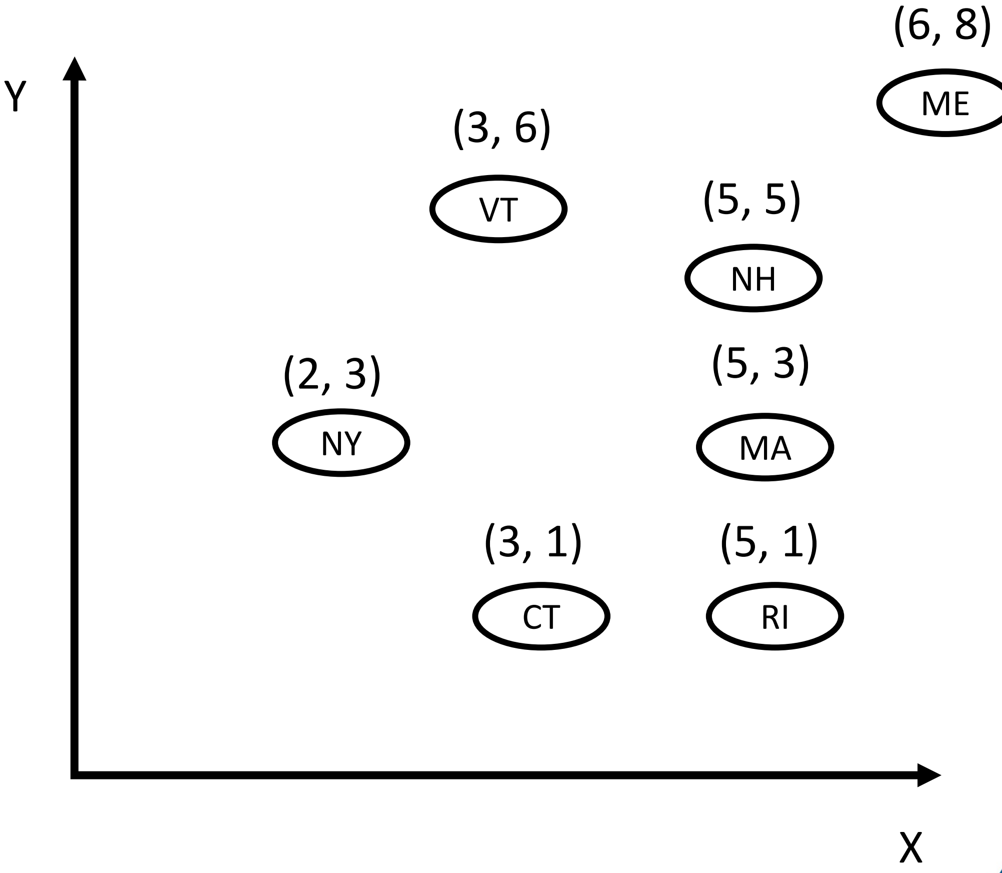
Attendance

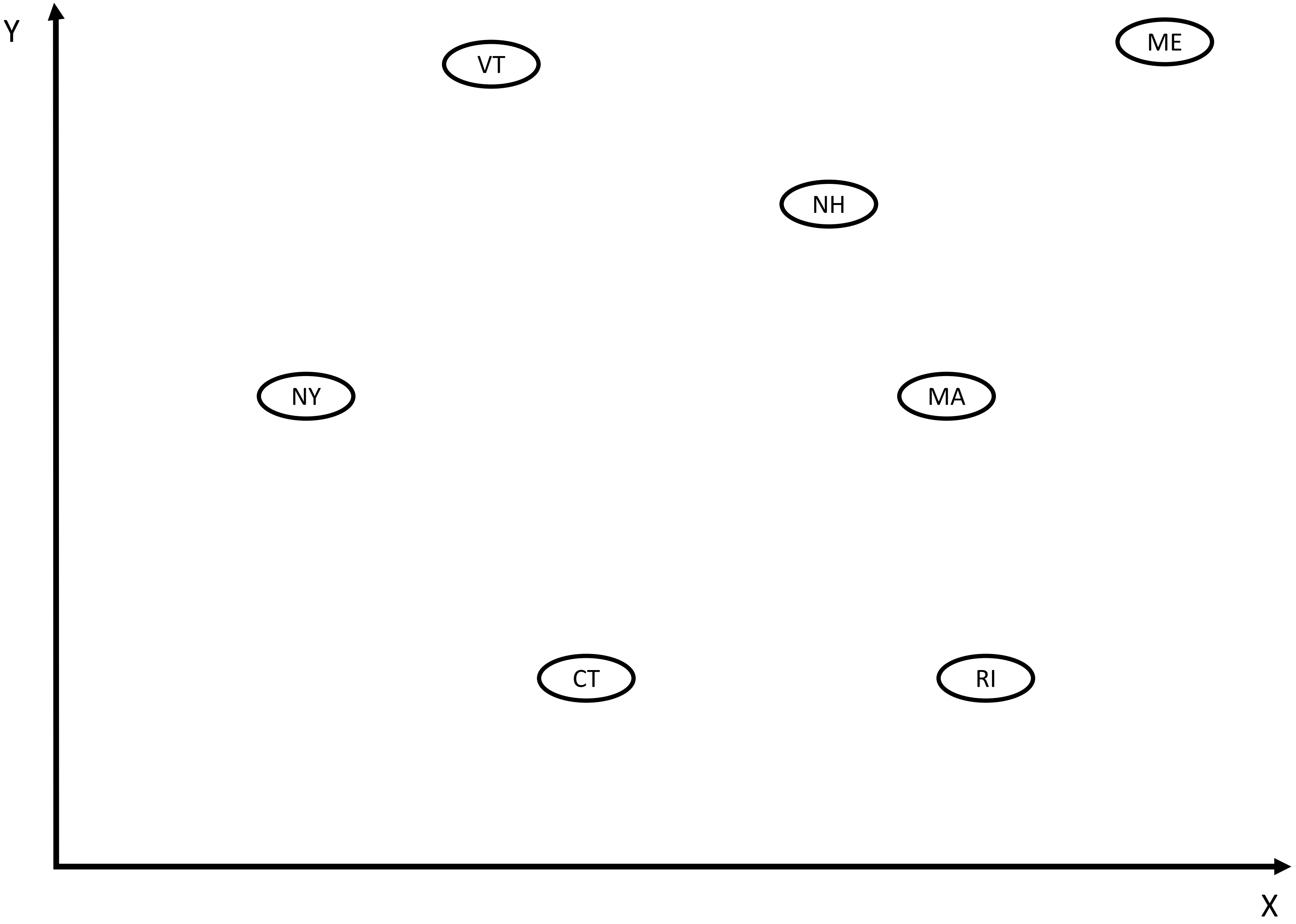
Today

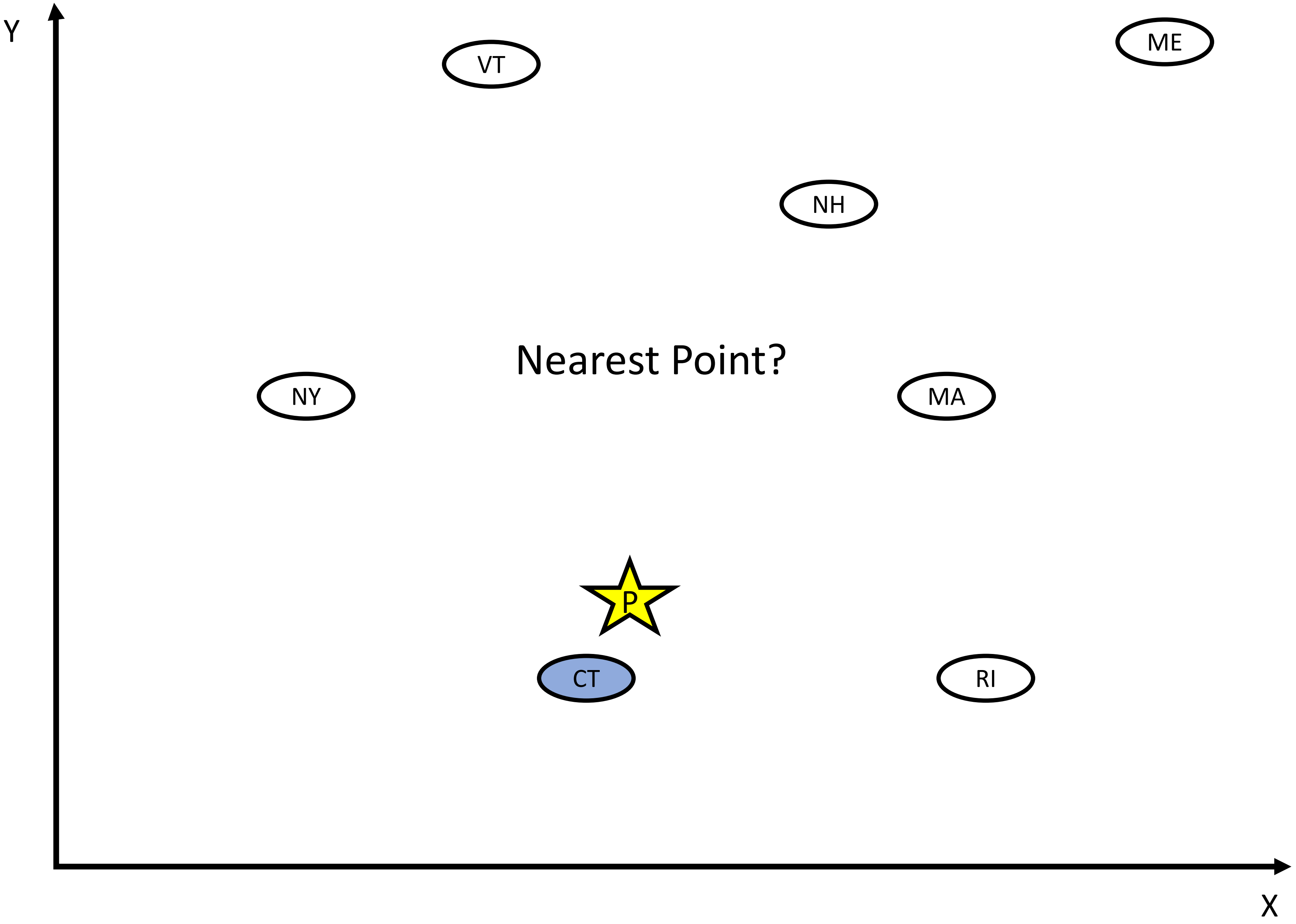
- kdTrees
- Spell Checker
- Exercise hints
- Finish through exercise 2
    - Show us passing Gradescope tests to leave early

KdTrees

- Symbol table implementation
- Stores Points from a plane
- A value is stored at each point
- Ex. Points on a map holding a string value denoting the state they're in

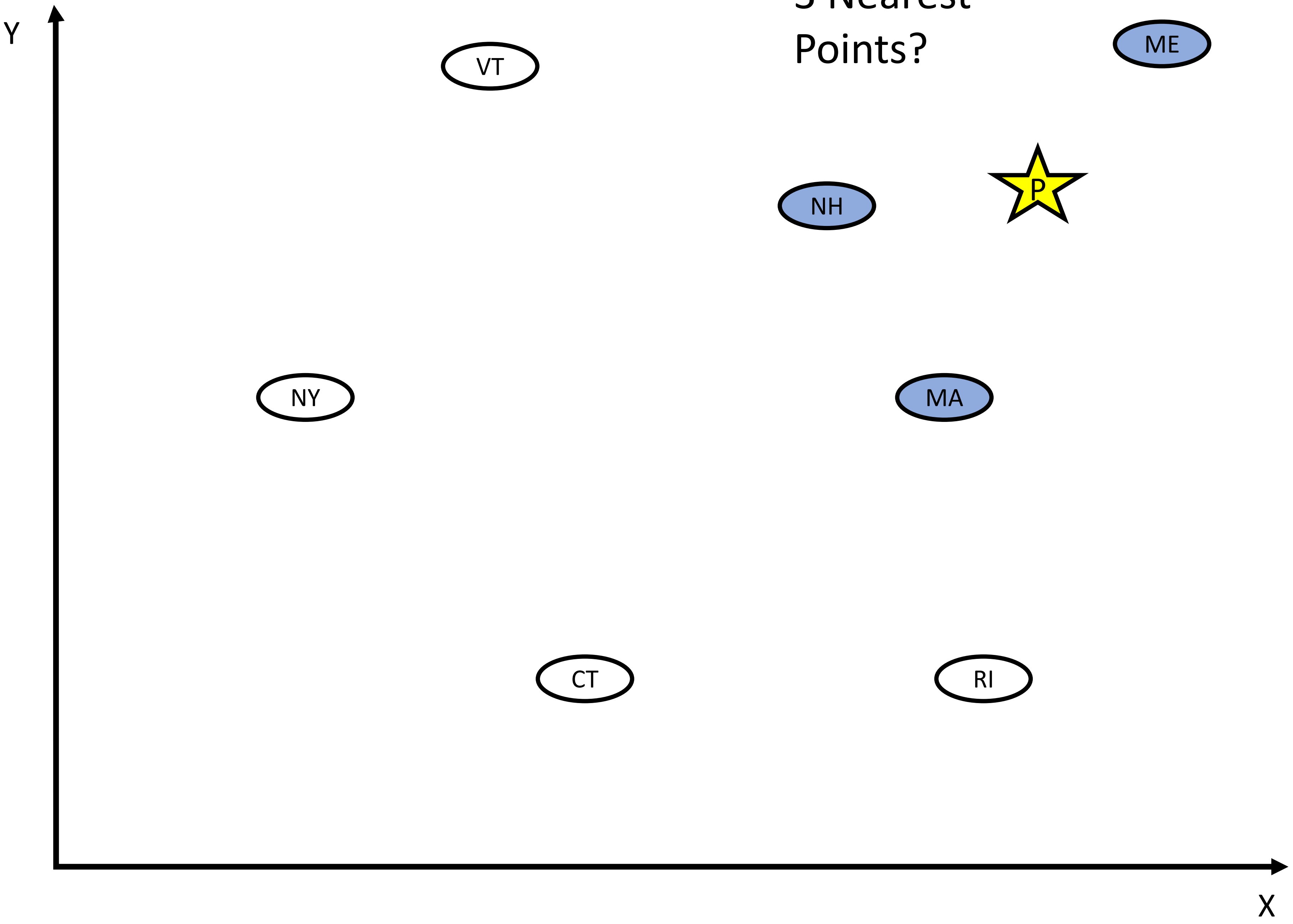


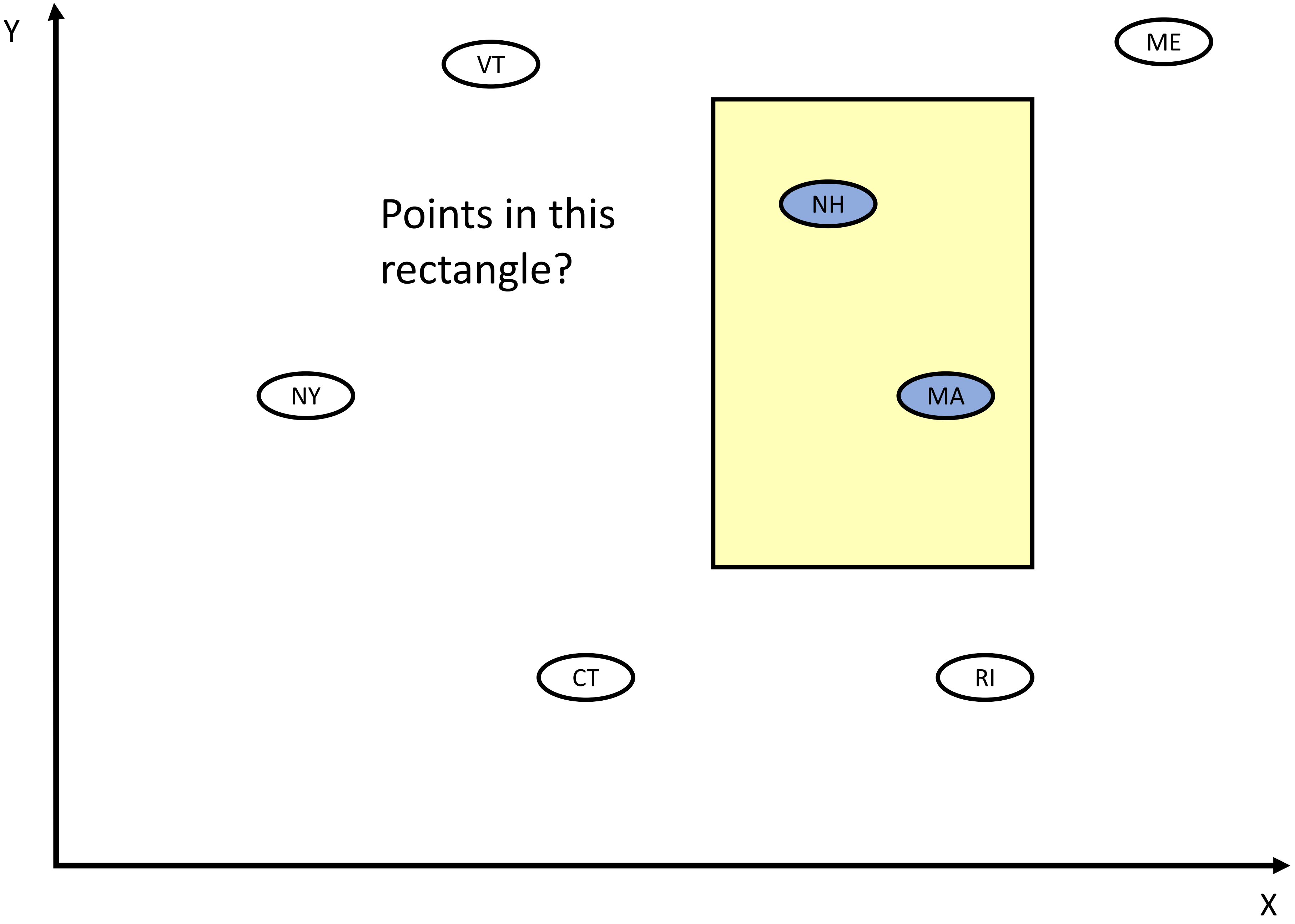

Y
VT
ME
NH
NY
MA
CT
RI
X



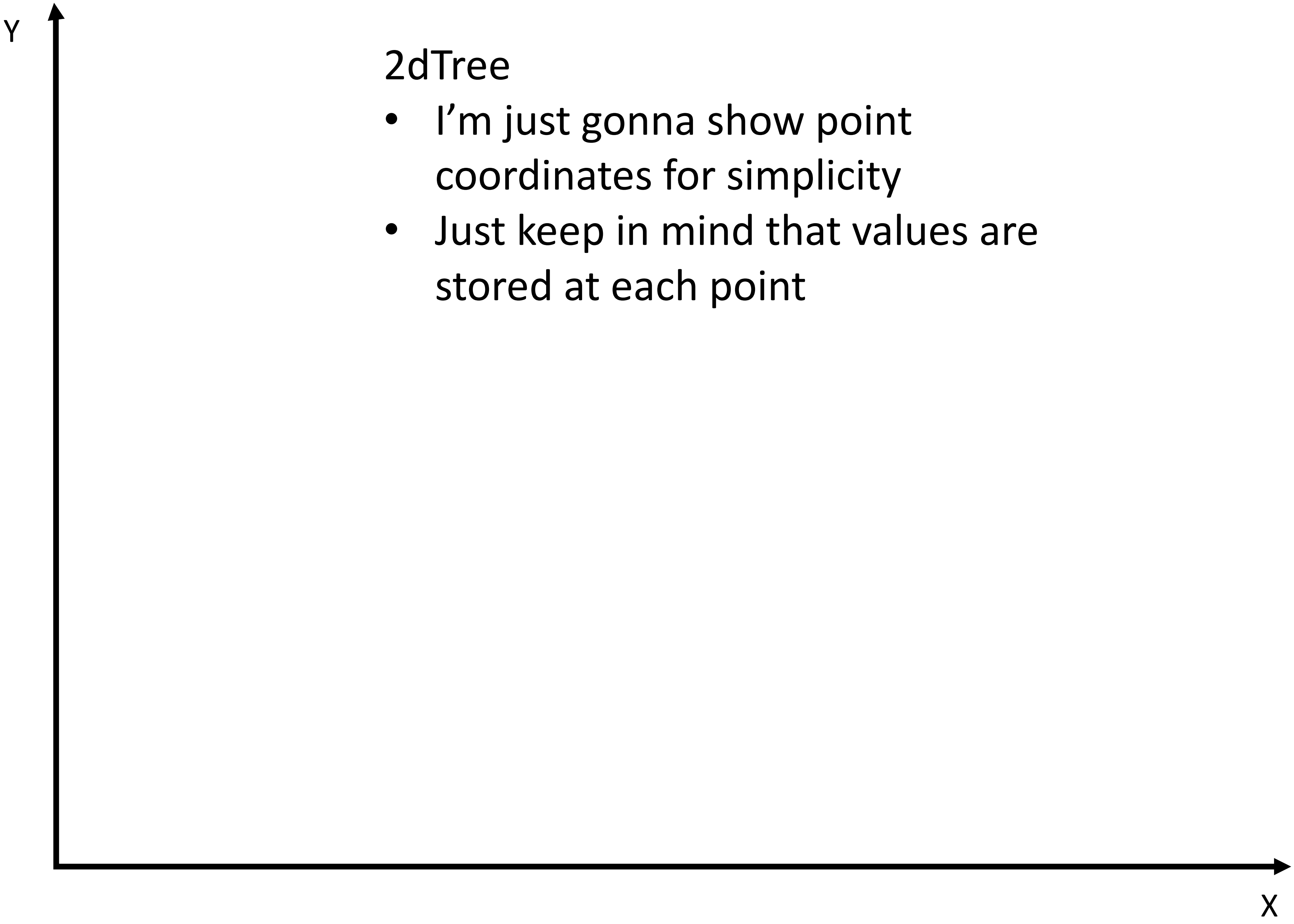
Nearest Point?

3 Nearest Points?





Points in this rectangle?

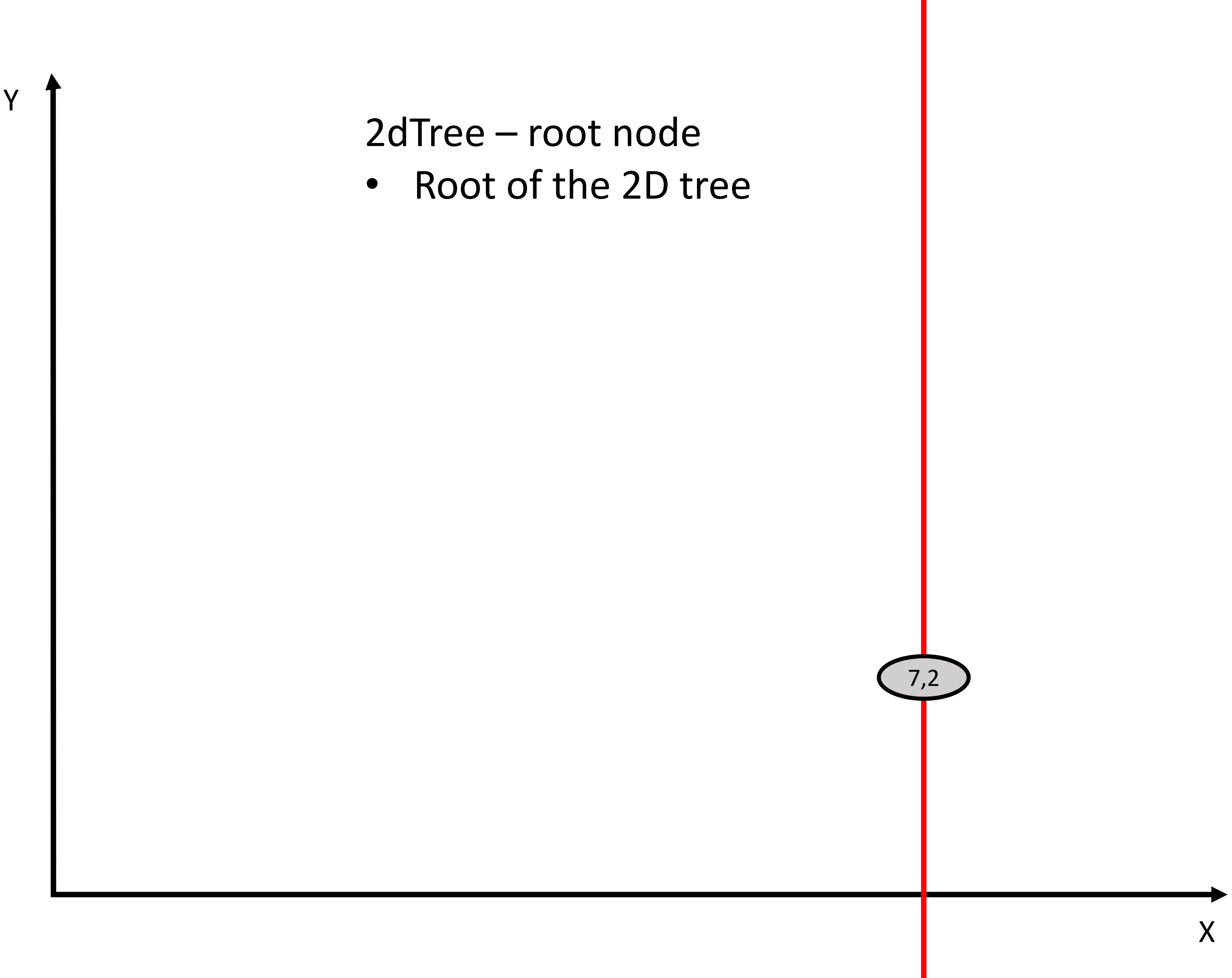


2dTree

- I’m just gonna show point coordinates for simplicity
- Just keep in mind that values are stored at each point

2dTree – root node

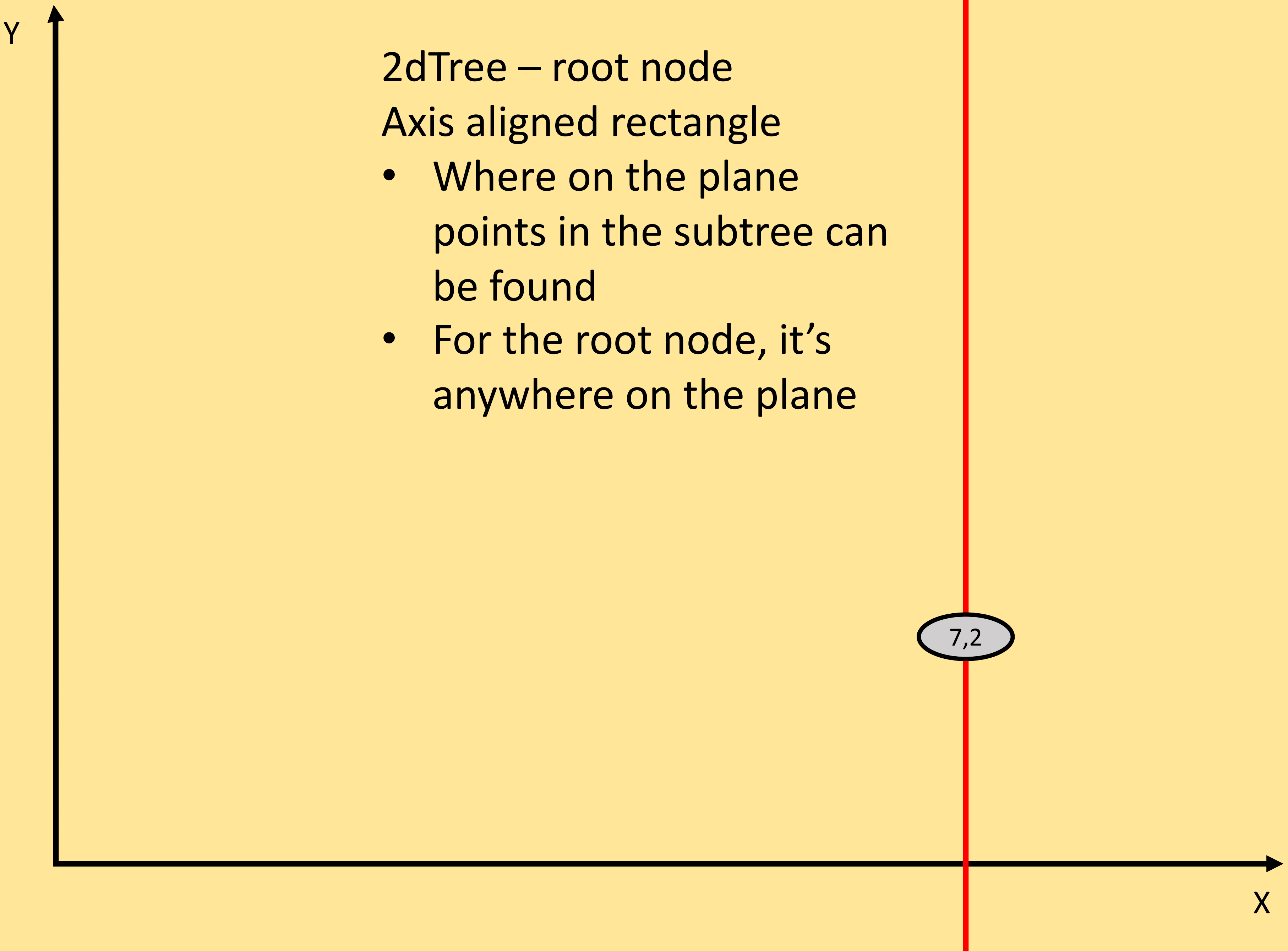
- Root of the 2D tree



2dTree – root node

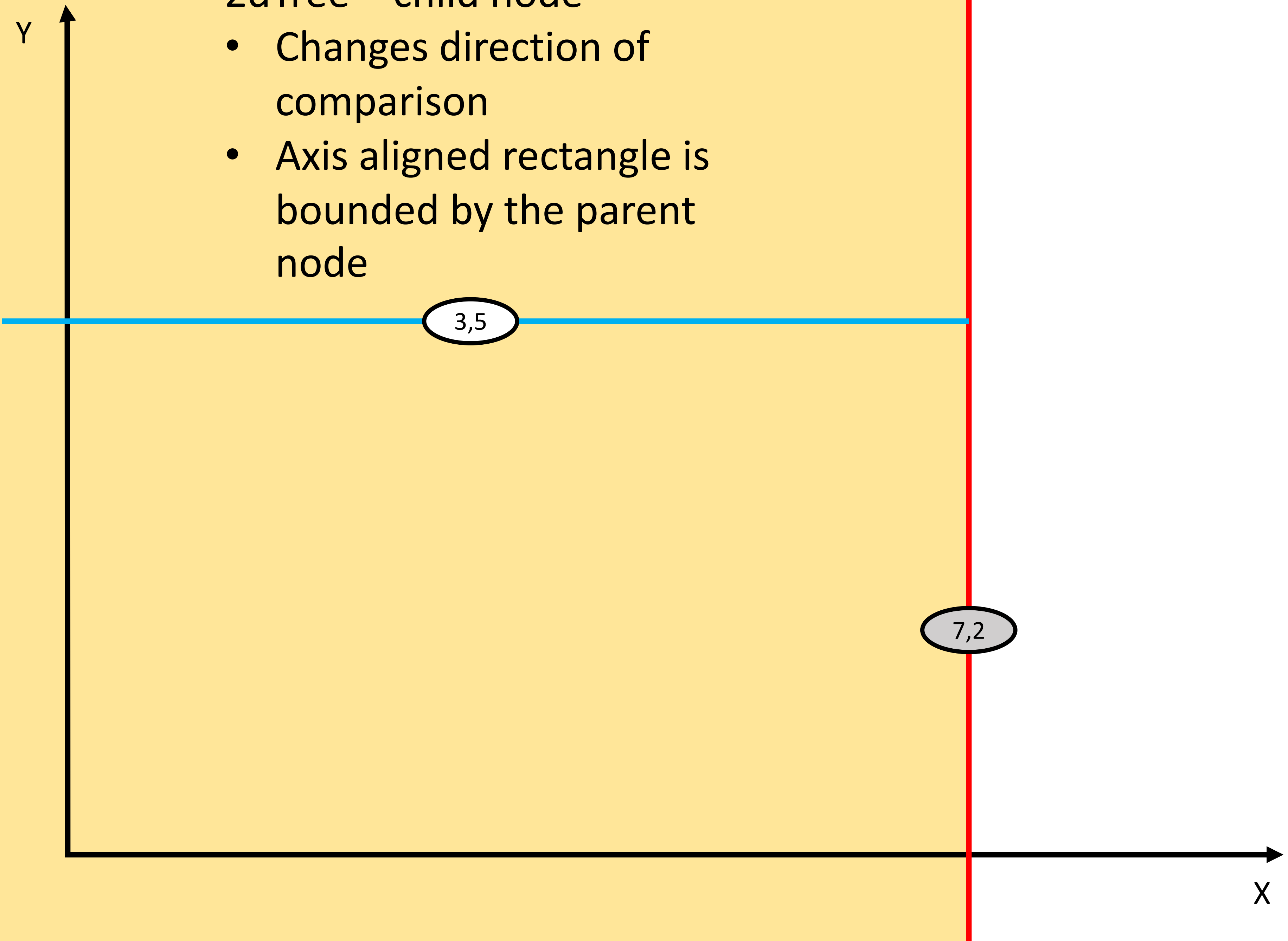
Axis aligned rectangle

- Where on the plane points in the subtree can be found
- For the root node, it's anywhere on the plane



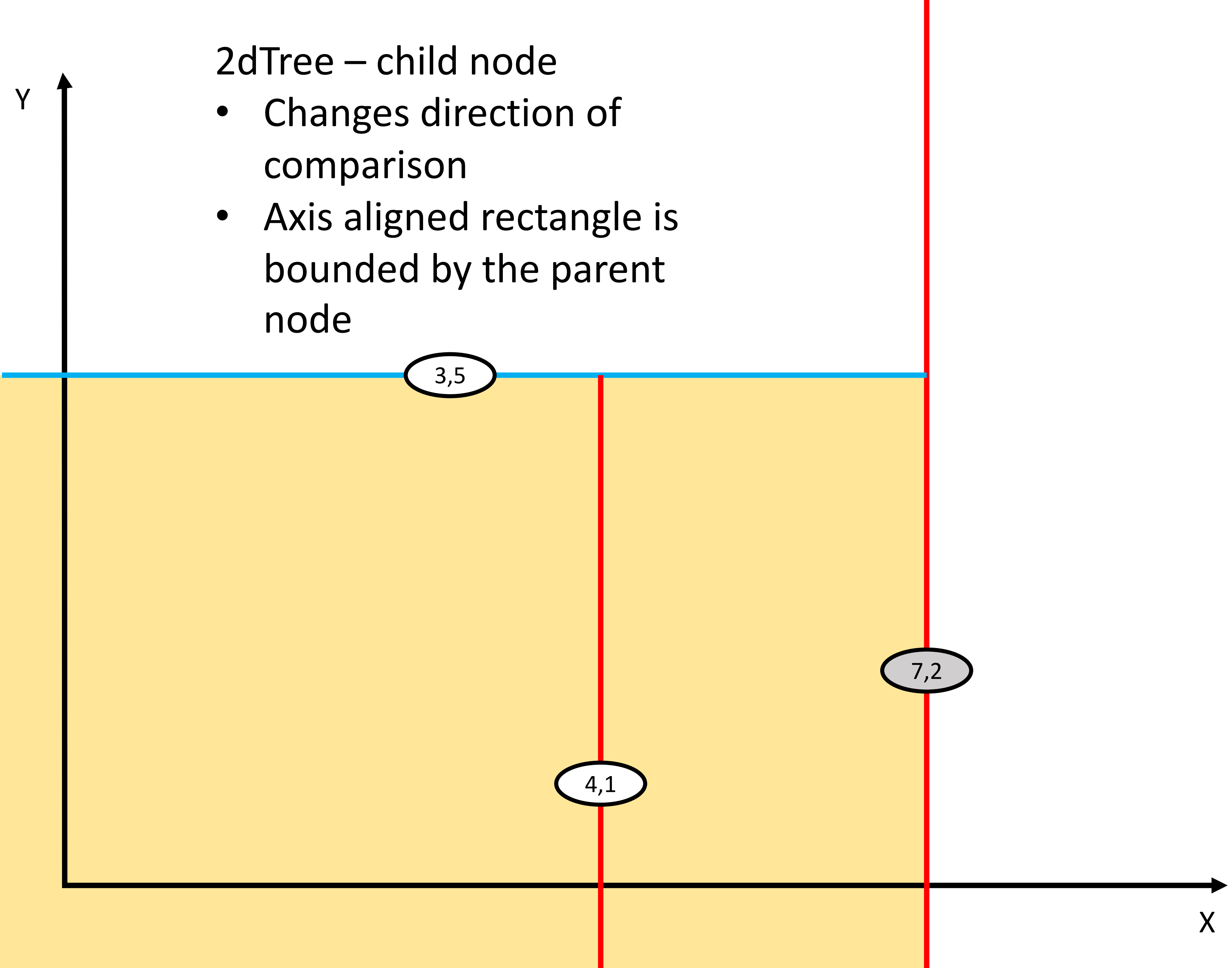
2dTree – child node

- Changes direction of comparison
- Axis aligned rectangle is bounded by the parent node



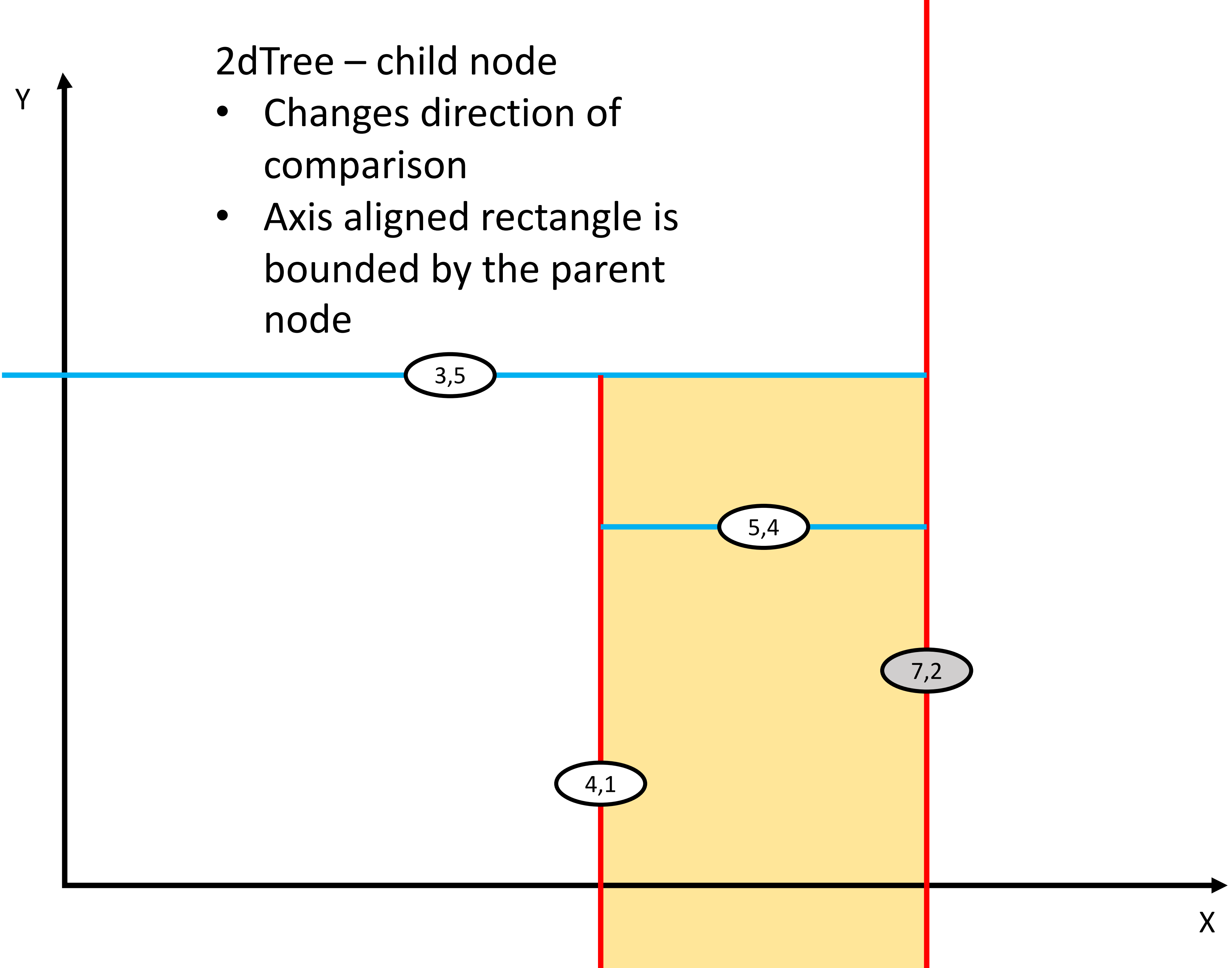
2dTree – child node

- Changes direction of comparison
- Axis aligned rectangle is bounded by the parent node

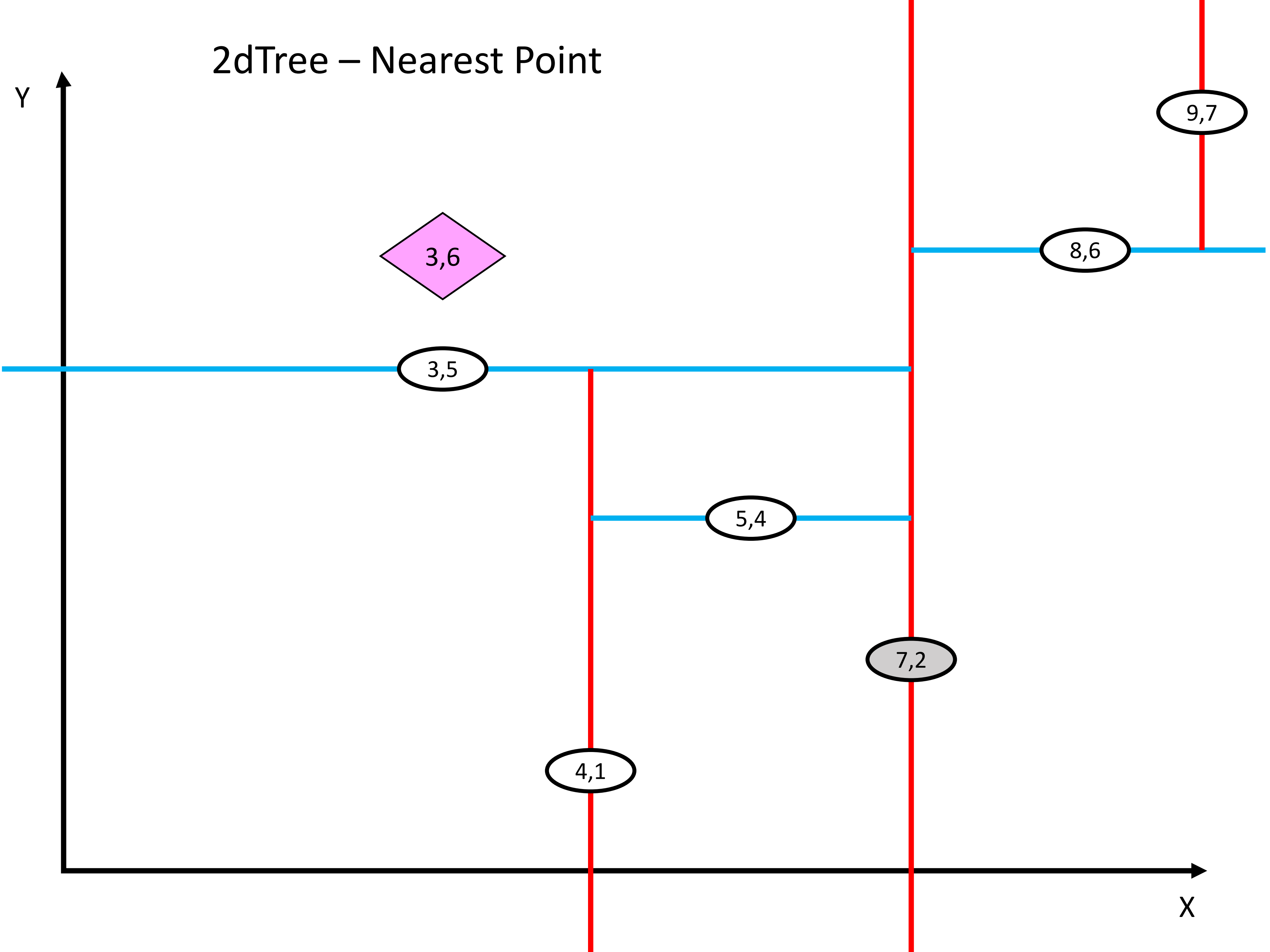


2dTree – child node

- Changes direction of comparison
- Axis aligned rectangle is bounded by the parent node

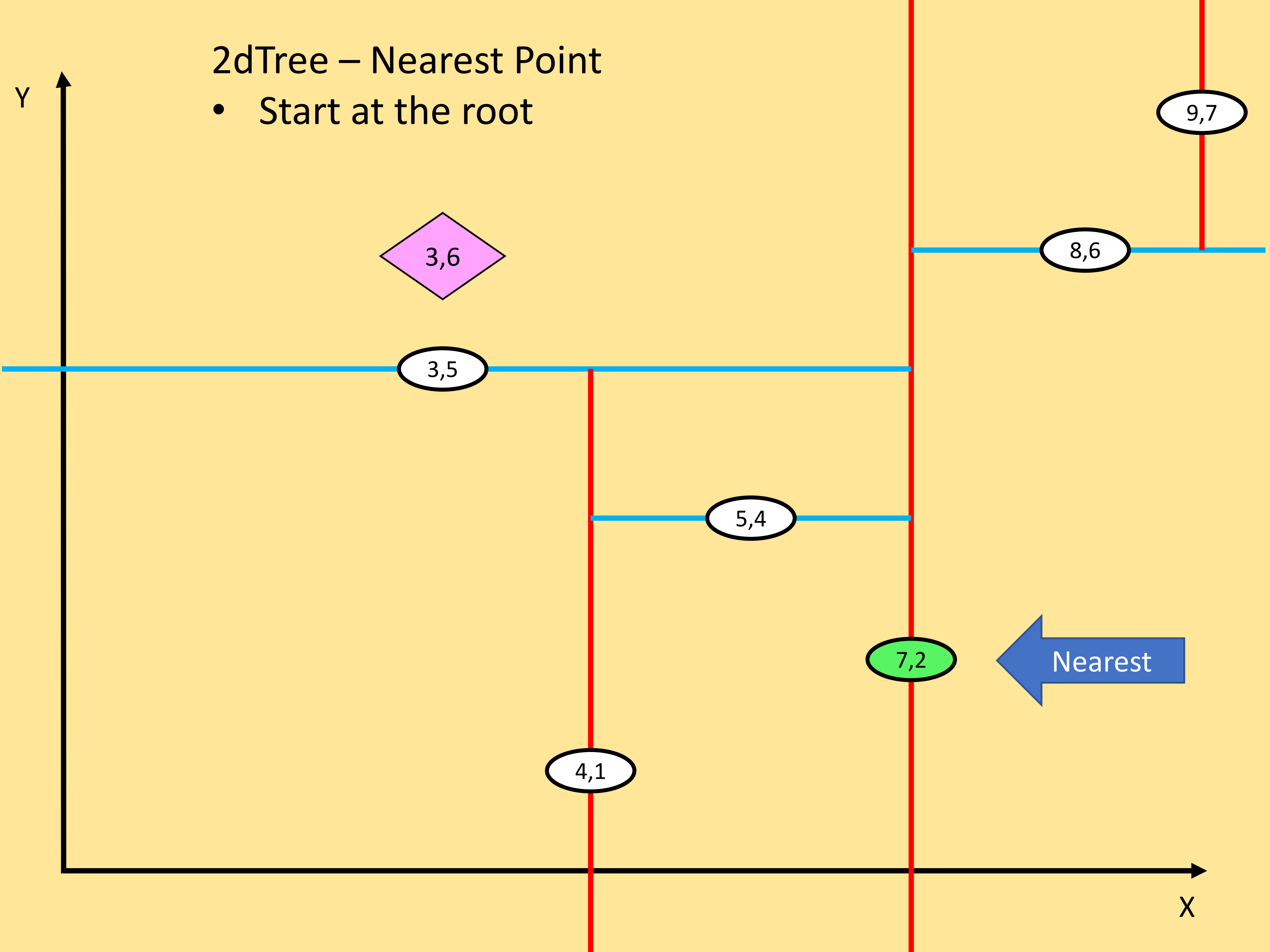


2dTree – Nearest Point



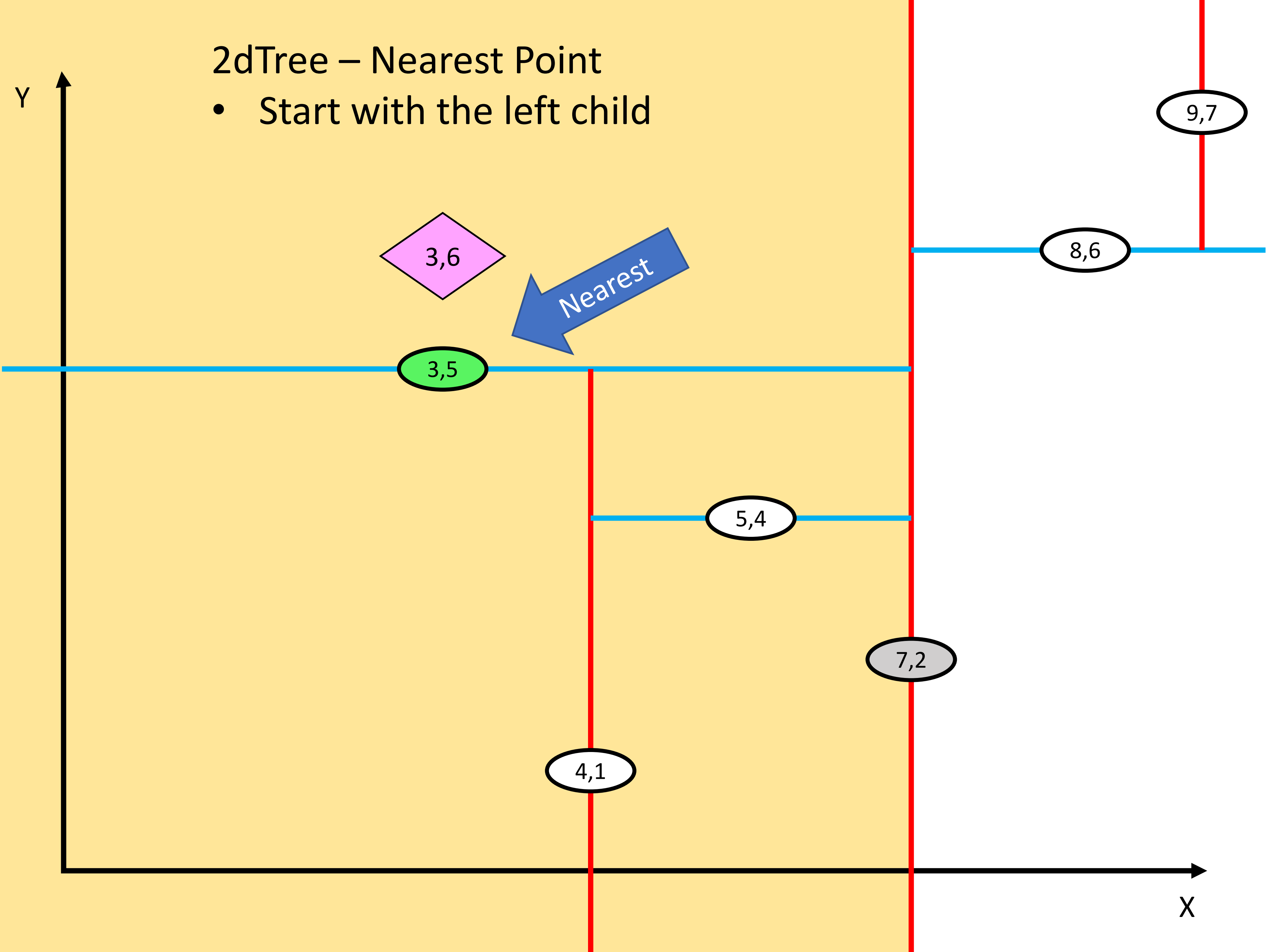
2dTree – Nearest Point

- Start at the root



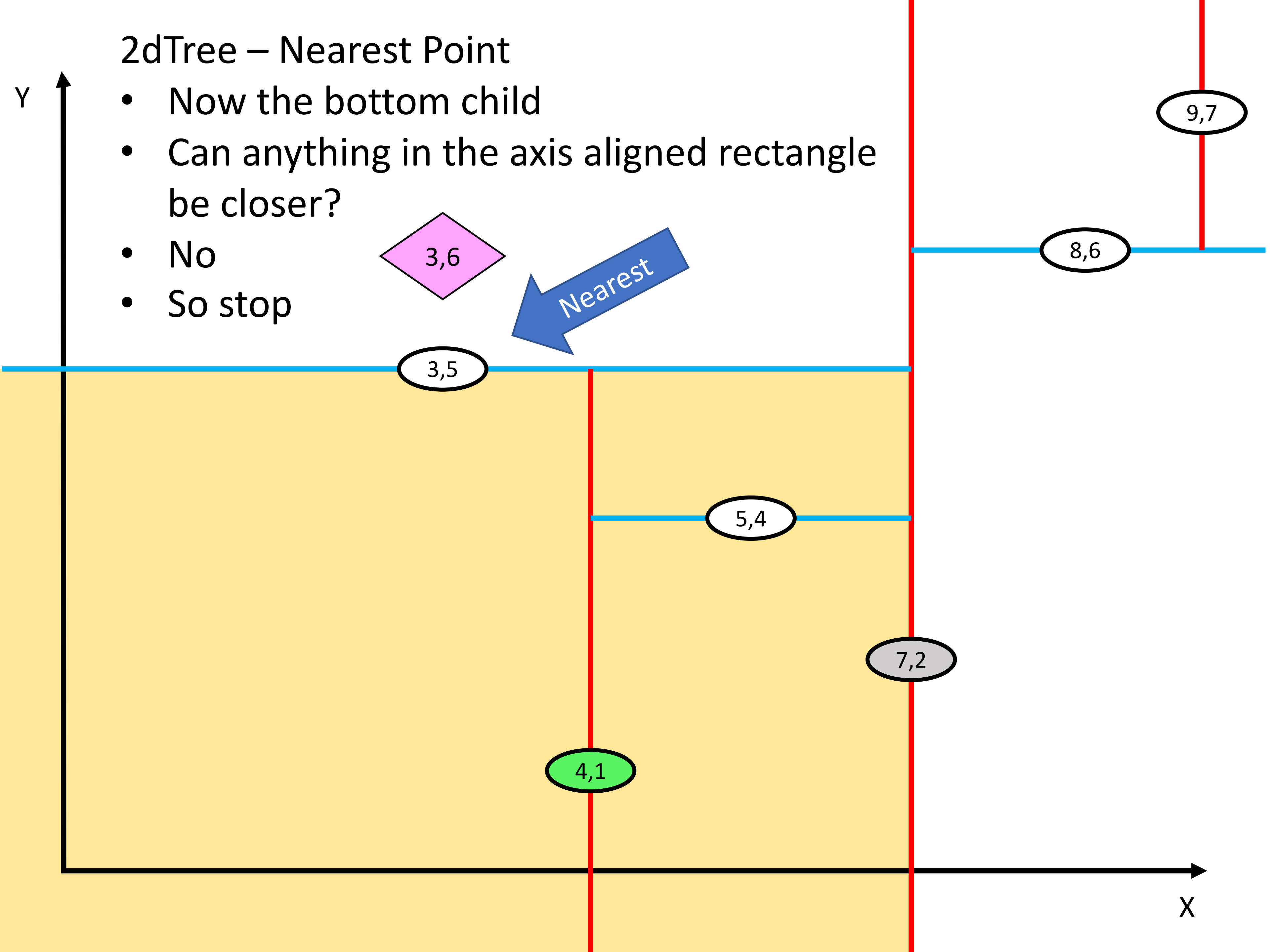
2dTree – Nearest Point

- Start with the left child



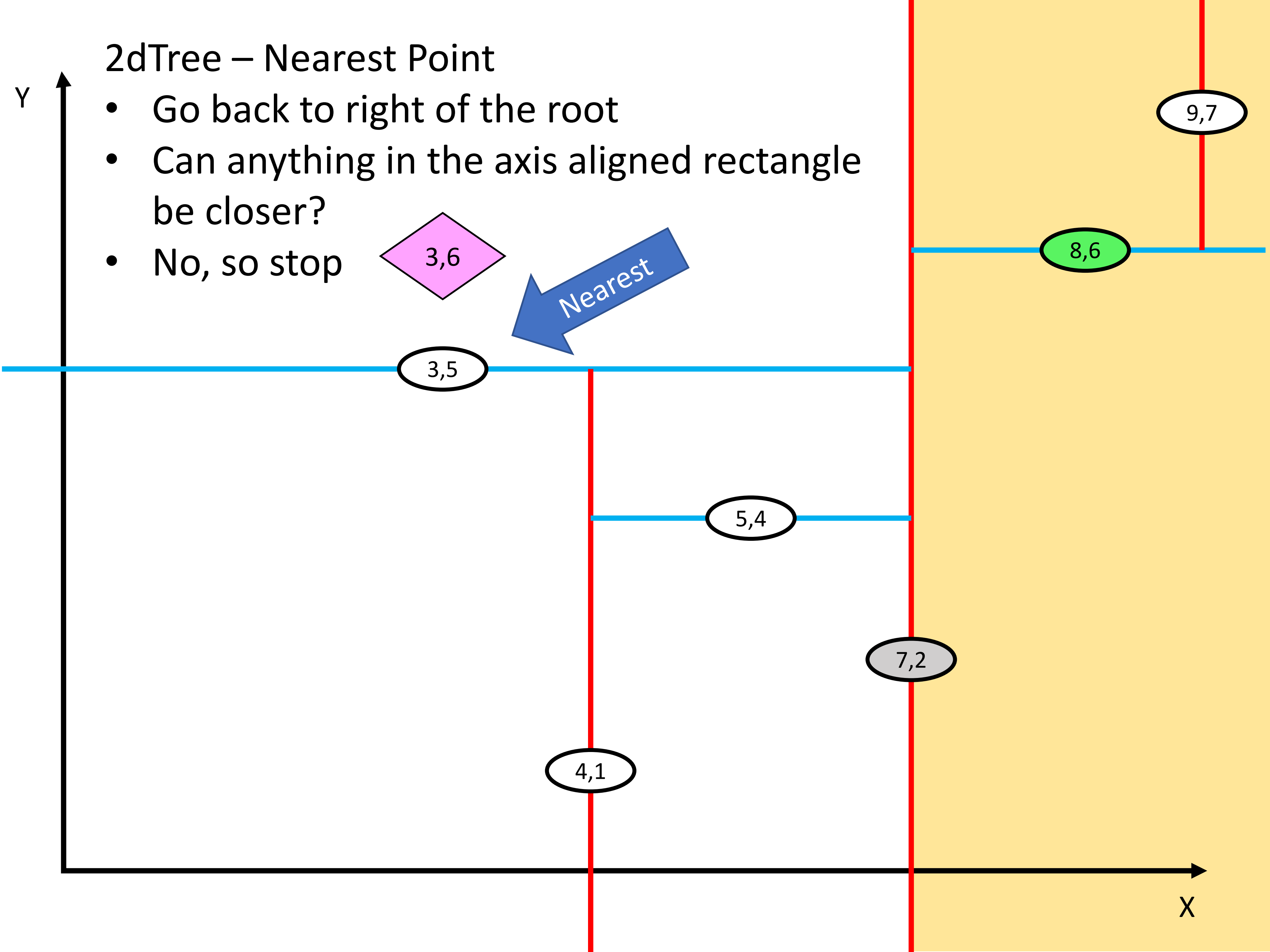
2dTree – Nearest Point

- Now the bottom child
- Can anything in the axis aligned rectangle be closer?
- No
- So stop



2dTree – Nearest Point

- Go back to right of the root
- Can anything in the axis aligned rectangle be closer?
- No, so stop

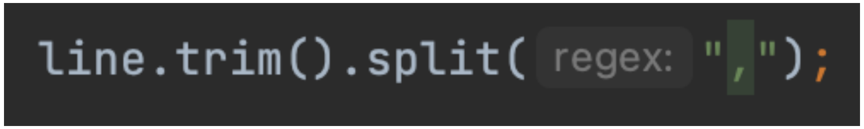


Questions?

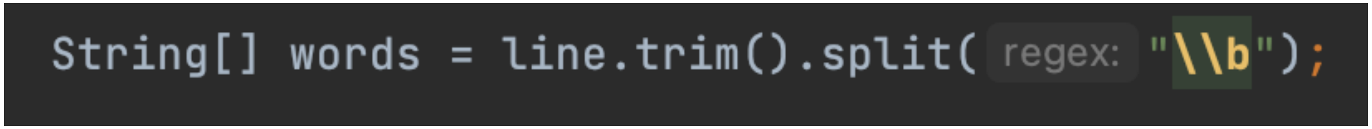
Spell Checker

Exercise hints - Spell Checker

- Splitting lines
  - Using “,”
  - Using “\\b”

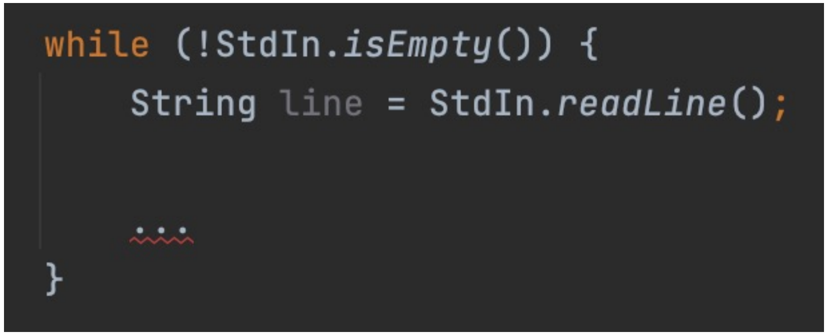


```
line.trim().split( regex: ",");
```



```
String[] words = line.trim().split( regex: "\\b");
```

- Reading StdIn
  - Lines
  - While loop

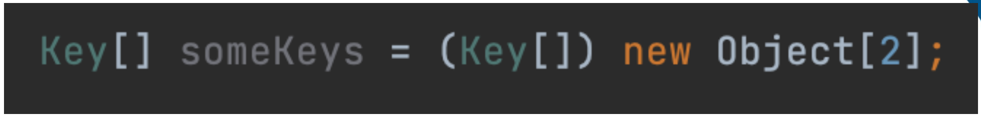


```
while (!StdIn.isEmpty()) {
    String line = StdIn.readLine();


    ...
}
```

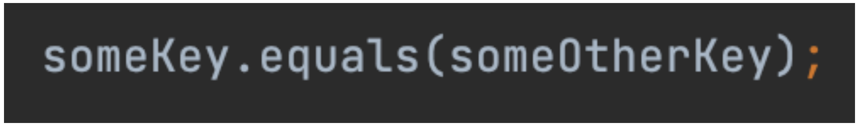
Exercise Hints

- Creating a generic array



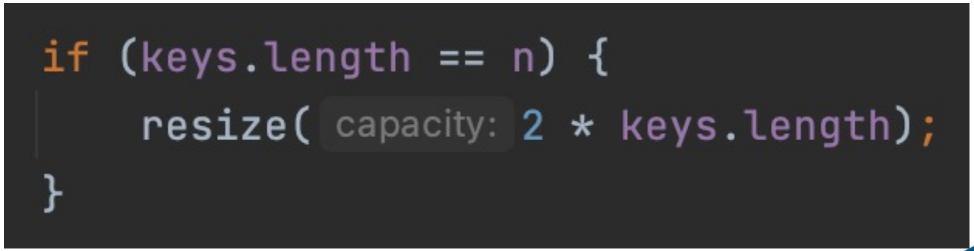
```java
Key[] someKeys = (Key[]) new Object[2];
```

- It’s an unordered array, so new entries can go at the end
- Shift key/value pairs to the left to delete a key
- Linear search to find entries (simple loop)
- Comparing Keys



```java
someKey.equals(someOtherKey);
```

- Keep track of n
- Resize arrays as necessary



```java
if (keys.length == n) {
    resize( capacity: 2 * keys.length);
}
```