Assume you need to record the number of people in each age for a population. Extend the BinarySearchTree class of your book as **AgeSearchTree** class. You will implement **AgeData** class to handle both age and number of people at that age values. You should keep instances of AgeData in your tree. Note that AgeData should be Comparable. CompareTo method of AgeData class will be used and the comparison will be done considering the age only. For the AgeSearchTree class you will override add, remove and find methods as follows:

- While adding a node, the **add** function will first check if a node with that age exists. If it exists, the number of people field of the AgeData object in that node will be increased 1. Otherwise a new node with the AgeData object will be inserted.
- While removing a node, the **remove** function will first check if a node with that age exists. If it exists and the number of people field of this node's AgeData object is greater than 1, it will decrease the number of people field 1. If the number of people field is 1, it will remove the node.
- The **find** method will get an AgeData object of any age and find the AgeData object with the same age and return it.
- Add a **youngerThan** method which returns the number of people younger than an age.
- Add an **olderThan** method which returns the number of people older than an age. Be careful! If your youngerthan and olderThan methods always traverse all nodes that you cannot get whole credit. You should traverse only the nodes needs to be traversed.

Here are some example code lines to help you:

```
//Create an empty age tree
AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();

//Add nodes for some ages, remove method works similarly
ageTree.add(new AgeData(10));
ageTree.add(new AgeData(20));
ageTree.add(new AgeData(5));
ageTree.add(new AgeData(15));
ageTree.add(new AgeData(10));

//Create a string representing the tree and print it treeStr =
ageTree.toString(); System.out.println(treeStr);
//This will print:
//10 - 2
//5 – 1
//null
//null
//20 - 1
//15 - 1
//null
//null
//null

//Print the number of people younger than 15
System.out.println(ageTree.youngerThan(15))
//output will be 3

//Find the number of people at any age System.out.println(ageTree.find(new
AgeData(10)).toString())
//It will print:
//10 - 2
```