BinaryCardTree.java BinaryCardTreeDriver.java BinaryCountNode.java Goals.java BinaryCardTree.txt

You will create a binary node class called BinaryCountNode that will be able to store in the node a Comparable and the number of times that node is in the tree. You will write a BinaryCardTree class that will store this node in a binary search tree. In this tree, if a node matches an already existing node, it will increase the count of the BinaryCountNode. Also, when removing any specific node, you must exhaust the count to zero before you remove the node from the tree, if it is not exhausted to zero, then it does not remove the node. The BinaryCardTree should have the following methods:

getAllCards which will return an ArrayList out all the Comparables in the tree the number of times it appears in the tree, such that if there is 4 Comparables called red, then it will print out red, red, red, red.

numCardsInHand which will retur the total number of cards in the tree.

getMostColorCard and get LeastColorCard will return an ArrayList of the cards of the most number and the cards that are the least color (there must be at least 1 of those cards.

removeCards which will return an ArrayList of cards that fulfill the number and Comparable requirements.

contains which will return the number of a specific Comparable in the tree

You will also write a small class called Goal which will store a number and a Comparable. This will be used as a goal to achieve in your hand.

the BinaryCardTree will hold a String representing a color of a card. Your goal is to draw a provided number into your hand (the binary search tree). Then you will read in a known list of Goals. Starting with the first goal, you will either remove the required cards from your hand OR, failing to do that, you will add 2 cards from a given deck. The deck is given after the list of goals. You will finish your program when either the Goal deck is finished or the deck you are drawing from is empty.

One comparable is called “wild”, this means it can be any color you need it to be at any given time.

Input File: The input file will start with a number, the initial number of cards in your hand, followed by those cards. Then there will be another number indicating the number of goals to read in the form of int followed by Comparable. The remaining elements is the deck.

Sample File :

```
10
wild
pink
```

```
yellow
black
white
white
blue
pink
green
wild
3
1 black
2 green
5 white
wild
orange
yellow
red
orange
green
black
blue
```

Sample Output :

```
hand size: 10
hand -- > [black, blue, green, pink, pink, white, white, wild, wild, yellow]
1-black is completed: [black]
2-green is completed: [green, wild]
Uncompleted Goals:
5-white
hand size: 15
hand -- > [black, blue, blue, green, orange, orange, pink, pink, red, white,
white, wild, wild, yellow, yellow]//same line
Color with most cards: [orange, blue, white, pink, yellow, wild]
Color with least cards: [black, green, red]
```