

Q2-)Test Case.Frequencies are taken from chapter 6, figure 6.37. b 13 ,c 22,d 32, a 64, e 103.

HW6 - [C:\Users\ORHAN\Documents\NetBeansProjects\HW6] - [HW6] - ...src\Main.java - IntelliJ IDEA 2016.3.6

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW6 > src > Main

Project: HW6 C:\Users\ORHAN\Documents\NetBeansProjects\HW6

- .idea
- out
- Report
- src
 - BinaryTree
 - HuffmanTree
 - Main
- freq.txt
- HW6.iml
- External Libraries

Main.java

```
27 } catch (IOException e) {
28     e.printStackTrace();
29 }
30 HuffmanTree.HuffData[] hufdataArray=new HuffmanTree.HuffData[hufp
31
32 for(int i=0;i<hufpuf.size();++i){
33     hufdataArray[i]=hufpuf.get(i);
34 }
35
36
37 customHufPuf.buildTree(hufdataArray);
38
39 String en = customHufPuf.encode(lineToCode: "cadde");
40
41 String de=customHufPuf.decode(en);
42 System.out.println(de);
43
44
45
46
47 }
48
49
50
```

Run Main

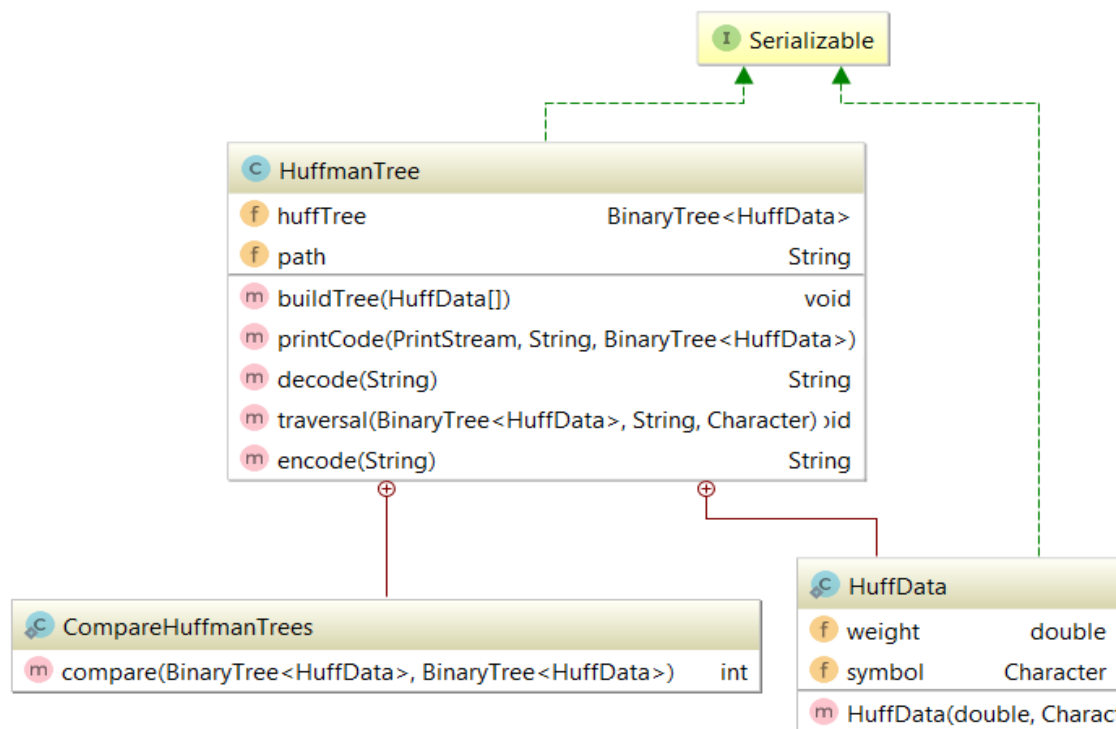
"C:\Program Files\Java\jdk1.7.0_80\bin\java" ...

1111
10
110
110
0
cadde

Process finished with exit code 0

Compilation completed successfully in 1s 689ms (3 minutes ago)

Class Diagram of HuffmanTree



Problem Solution Approach

I have used global string to assign the path when the char has found. I have modified recursive traversal method.

```

private String path;

protected void traversal(BinaryTree<HuffData> rootNode ,String yol,Character key ) {

    if(rootNode!=null) {
        if(rootNode.getData().symbol==key) {
            path=yol;
            return;
        }

        String temp=yol;
        traversal(rootNode.getLeftSubtree(), temp+"0", key);
        temp=yol;
        traversal(rootNode.getRightSubtree(), temp+"1", key);
    }
}
    
```

Then i used this method in my encode method.

```
public String encode(String lineToCode){
    StringBuilder result = new StringBuilder();
    BinaryTree < HuffData > currentTree = huffTree;
    for (int i=0;i<lineToCode.length();i++){
        String yol="";
        traversal(currentTree,yol,lineToCode.charAt(i));
        result.append(path);
        System.out.println(path+" ");////Print line.It's okay to remove.
    }
    return result.toString();
}
```

Q3-)Test Case.

```
public class Main {

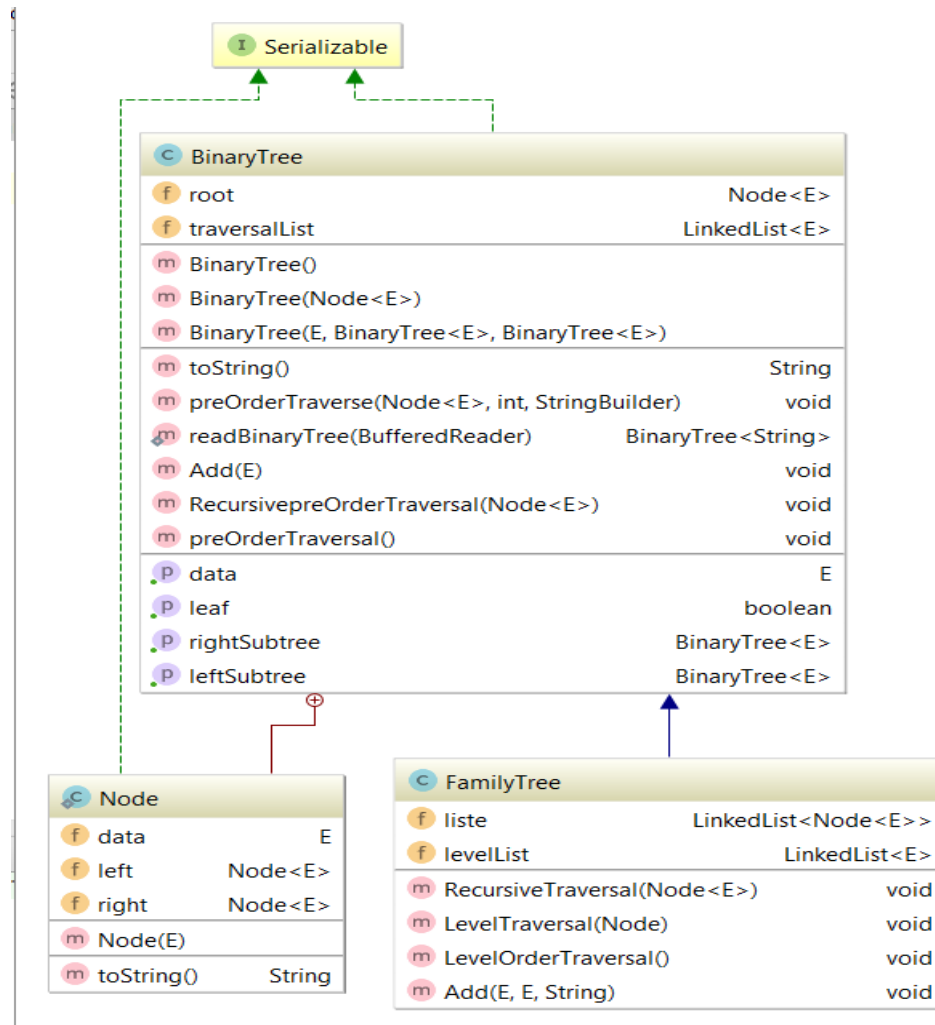
    public static void main(String[] args) {

        FamilyTree<String> family= new FamilyTree<>();
        try {
            family.Add( person: "Hasan", Parent: null, nick: null);
            family.Add( person: "Ayşe", Parent: "Hasan", nick: "ebu-Ayşe");
            family.Add( person: "Ali", Parent: "Ayşe", nick: "ibn-Hasan");
            family.Add( person: "Sema", Parent: "Hasan", nick: "ebu-Ayşe");
        } catch (Exception e) {
            e.printStackTrace();
        }

        family.LevelOrderTraversal();

    }
}
```

Class Diagram.



Problem Solution Approach.

LevelList is a global variable of the class. I have used FIFO approach to put subtrees in a linked list. Idea is getting the leaves of the first root that has entered the linked list. First left then right.

```
protected LinkedList<E> levelList = new LinkedList<>();
```

```
private void LevelTraversal(Node root){
    LinkedList<Node> level = new LinkedList<>();
    level.add(root);
    while(!level.isEmpty()){
        Node node = level.poll();

        levelList.add((E) node.data);

        if(node.left!= null)
            level.add(node.left);
        if(node.right!= null)
            level.add(node.right);
    }
}
```

Then use these methods for our main method. That's why other method is private.

```
public void LevelOrderTraversal(){//////////HW 6 Question 3

    LevelTraversal(super.root);

    Iterator<E> it=levelList.iterator();

    while (it.hasNext())
        System.out.println(it.next());

    levelList.clear();
}
```

It wasn't clear in the whether we were just print all of it or return them as string or return the method as an iterator. So I have used an iterator in the. If necessary I can easily change the method into an iterator.