

Cheewin Thawornjaroenpong

CSCI323.25 Designs and Analysis of Algorithms (Spring 2023)

Project3

Huffman coding scheme

02/27/2023

Algorithm Steps:

Step 0: inFile open from args [0] outFile1, deBugFile open from args[1], args[2]
 Step 1: listHead get a HNode ("dummy", 0, "", 0, 0, null, null, null) as the dummy node // use constructor
 Step 2: constructHuffmanLLList (inFile, deBugFile, listHead)
 Step 2: printList (listHead, outFile1)
 Step 3: Root constructHuffmanBinTree (listHead, deBugFile)
 Step 4: outFile1 "printing the entropy Table"
 Step 5: totalEntropy 0
 Step 6: printEntropyTable (Root, "", outFile1, totalEntropy) // "" is an empty string
 Step 7: outFile1 "The Huffman Coding scheme's entropy = " (double) totalEntropy / 100.00
 Step 8: preOrder (Root, outFile1)
 Step 9: inOrder (Root, outFile1)
 Step 10: postOrder (Root, outFile1) Step 11: close all files

Illustrations:**Source code:**

```
import java.io.*;
import java.util.*;
public class ThawornjaroenpongC_Project3_Main {

    public static void main(String[] args) {
        try {
            Scanner inFile = new Scanner(new FileReader(args[0]));
            BufferedWriter outFile = new BufferedWriter(new FileWriter(args[1]));
            BufferedWriter deBugFile = new BufferedWriter(new FileWriter(args[2]));

            HNode listHead = new HNode("dummy", 0, "", 0, 0, null, null, null);
            Huffman HM = new Huffman();
            HM.constructHuffmanLLList(inFile, deBugFile, listHead);
            HM.printList(listHead, outFile);

            HNode Root = HM.constructHuffmanBinTree(listHead, deBugFile);
            outFile.write("printing the entropy Table \n");
            double totalEntropy = 0;
            HM.printEntropyTable(Root, "", outFile, totalEntropy);
            outFile.write("The Huffman Coding scheme's entropy = " + HM.totalEntropy +
"\n");

            outFile.write("*****PreOrder Traversal***** \n");
            HM.preOrder(Root, outFile);
            outFile.write("*****InOrder Traversal***** \n");
            HM.inOrder(Root, outFile);
```

```

        outFile.write("*****PostOrder Traversal***** \n");
        HM.postOrder(Root, outFile);

        inFile.close();
    outFile.close();
    debugFile.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

static class HNode
{
    private String chStr;
    private int prob;
    private String code;
    private int numBits;// string length of code
    private double entropy;// numBits * prob
    private HNode left;
    private HNode right;
    private HNode next;

    HNode(String ch, int prob, String code,int bits, double entropy, HNode left,
HNode right, HNode next)
    {
        this.chStr = ch;
        this.prob = prob;
        this.code = code;
        this.numBits = bits;
        this.entropy = entropy;
        this.left = left;
        this.right = right;
        this.next = next;
    }

    public void printNode(HNode T, BufferedWriter outFile)
    {
        try {

            outFile.write("T's chStr is " + T.chStr + " T's prob is " +
T.prob +

```

```

T.numBits + "\n" +
        " T's code is " + T.code + " T's numBits is " +
        " T's entropy is " + T.entropy);
    if(T.next == null)
    {
        outFile.write(" next's chStr is null ");
    }
    else
    {
        outFile.write(" next's chStr is " + T.next.chStr );
    }
    if(T.left == null)
    {
        outFile.write(" left's chStr is null ");
    }
    else
    {
        outFile.write(" left's chStr is " + T.left.chStr);
    }
    if(T.right == null)
    {
        outFile.write(" right's chStr is null ");
    }
    else
    {
        outFile.write(" right's chStr is " + T.right.chStr);
    }

    outFile.write("\n");
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

static class Huffman
{
    private HNode listHead;
    private HNode Root;
    private int totalEntropy;

    Huffman()
    {

```

```

        this.listHead = null;
        this.Root = null;
        this.totalEntropy = 0;
    }
    public HNode findSpot(HNode listHead, HNode newNode)
    {
        HNode spot = listHead;

        while(spot.next != null && spot.next.prob < newNode.prob)
        {
            spot = spot.next;
        }

        return spot;
    }

    public void listInsert(HNode listHead, HNode newNode, BufferedWriter
debugFile)
    {
        try {
            debugFile.write("In listInsert method");
            debugFile.write("\n");
            HNode spot = this.findSpot(listHead, newNode);
            debugFile.write("Returns from findSpot where Spot.data is " +
spot.chStr);

            debugFile.write("\n");
            debugFile.write("newNode.data is " + newNode.chStr);
            debugFile.write("\n");

            newNode.next = spot.next;
            spot.next = newNode;
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public void constructHuffmanLList(Scanner inFile, BufferedWriter debugFile,
HNode listHead)
    {
        try {
            debugFile.write("Entering constructHuffmanLList method \n");
            String chr = "";

```

```

        int prob = 0;
        while(inFile.hasNext())
        {
            chr = String.valueOf(inFile.next().charAt(0)) ;
            prob = Integer.parseInt(inFile.next());
            HNode newNode = new HNode(chr, prob, "", 0, 0.0, null,
null, null);

            this.listInsert(listHead, newNode, debugFile);

            this.printList(listHead, debugFile);
        }

        debugFile.write("Exit constructHuffmanLList method");

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public HNode constructHuffmanBinTree(HNode listHead, BufferedWriter
debugFile)
{
    HNode tmp = listHead;
    try {
        debugFile.write("Entering        constructHuffmanBinTree
method \n");

        while(tmp.next.next != null)
        {
            HNode newNode = new HNode("", 0, "", 0, 0, null,
null, null);

            newNode.prob      =      tmp.next.prob      +
tmp.next.next.prob;

            newNode.chStr     =      tmp.next.chStr     +
tmp.next.next.chStr;

            newNode.left = tmp.next;
            newNode.right = tmp.next.next;
            newNode.next = null;
            this.listInsert(listHead, newNode, debugFile);
            tmp.next = tmp.next.next.next;
            this.printList(this.listHead, debugFile);
        }
    }
}

```

```

        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return tmp.next;
    }

    public void printEntropyTable(HNode T, String code, BufferedWriter outFile,
double totalEntropy)
    {
        try {

            if(T.left == null && T.right == null)
            {
                T.code = code;
                T.numBits = code.length();
                T.entropy = T.prob * T.numBits;
                this.totalEntropy += T.entropy;
                outFile.write("output T.chStr " + T.chStr + " T.prob "
+ T.prob + " T.code " +
                                T.code + " T.numBits " + T.numBits +
" T.entropy" + T.entropy + "\n");
            }
            else
            {
                printEntropyTable(T.left, code + "0", outFile,
totalEntropy+=T.entropy);
                printEntropyTable(T.right, code + "1", outFile,
totalEntropy+=T.entropy);
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    public void preOrder(HNode T, BufferedWriter outFile)
    {
        if(this.isLeaf(T))

```

```

        {
            T.printNode(T, outFile);
        }
        else
        {
            T.printNode(T, outFile);
            preOrder(T.left, outFile);
            preOrder(T.right, outFile);
        }
    }

public void inOrder(HNode T, BufferedWriter outFile)
{
    if(this.isLeaf(T))
    {
        T.printNode(T, outFile);
    }
    else
    {
        inOrder(T.left, outFile);
        T.printNode(T, outFile);
        inOrder(T.right, outFile);
    }
}

public void postOrder(HNode T, BufferedWriter outFile)
{
    if(this.isLeaf(T))
    {
        T.printNode(T, outFile);
    }
    else
    {
        postOrder(T.left, outFile);
        postOrder(T.right, outFile);
        T.printNode(T, outFile);
    }
}

public boolean isLeaf(HNode T)
{
    if(T.left == null && T.right == null)
    {
        return true;
    }
}

```



```

    }
    else
    {
        return false;
    }
}

```

```

public void printList(HNode listHead, BufferedWriter File) {
    try {
        int count = 0;
        HNode tmp = listHead;
        File.write("listHead" + "->" );
        while(tmp != null)
        {
            File.write("(");
            tmp.printNode(tmp, File);
            File.write(" -> " + "\n");
            tmp = tmp.next;
            count++;
            if(count >= 3)
            {
                File.write("\n");
            }
        }
        File.write("\n");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

}

```

```

}

```

Program output:

outFile from Data1 :

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

(T's chStr is d T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null
) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null
) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null
) ->

(T's chStr is g T's prob is 15 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

printing the entropy Table

output T.chStr a T.prob 40 T.code 0 T.numBits 1 T.entropy40.0
output T.chStr i T.prob 10 T.code 100 T.numBits 3 T.entropy30.0
output T.chStr o T.prob 5 T.code 1010 T.numBits 4 T.entropy20.0
output T.chStr d T.prob 5 T.code 10110 T.numBits 5 T.entropy25.0
output T.chStr h T.prob 5 T.code 10111 T.numBits 5 T.entropy25.0
output T.chStr g T.prob 15 T.code 110 T.numBits 3 T.entropy45.0
output T.chStr e T.prob 20 T.code 111 T.numBits 3 T.entropy60.0
The Huffman Coding scheme's entropy = 245

*****PreOrder Traversal*****

T's chStr is aiodhge T's prob is 100 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iodhge
T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1
T's entropy is 40.0 next's chStr is iodhge left's chStr is null right's chStr is null
T's chStr is iodhge T's prob is 60 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is aiodhge left's chStr is iodh right's chStr is ge
T's chStr is iodh T's prob is 25 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is ge left's chStr is i right's chStr is odh
T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
T's entropy is 30.0 next's chStr is odh left's chStr is null right's chStr is null
T's chStr is odh T's prob is 15 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is g left's chStr is o right's chStr is dh
T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
T's entropy is 20.0 next's chStr is dh left's chStr is null right's chStr is null
T's chStr is dh T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is i left's chStr is d right's chStr is h
T's chStr is d T's prob is 5 T's code is 10110 T's numBits is 5
T's entropy is 25.0 next's chStr is h left's chStr is null right's chStr is null
T's chStr is h T's prob is 5 T's code is 10111 T's numBits is 5
T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
T's chStr is ge T's prob is 35 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is a left's chStr is g right's chStr is e
T's chStr is g T's prob is 15 T's code is 110 T's numBits is 3
T's entropy is 45.0 next's chStr is e left's chStr is null right's chStr is null
T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
T's entropy is 60.0 next's chStr is iodh left's chStr is null right's chStr is null

*****InOrder Traversal*****

T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1
T's entropy is 40.0 next's chStr is iodhge left's chStr is null right's chStr is null
T's chStr is aiodhge T's prob is 100 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iodhge
T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
T's entropy is 30.0 next's chStr is odh left's chStr is null right's chStr is null
T's chStr is iodh T's prob is 25 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is ge left's chStr is i right's chStr is odh
T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
T's entropy is 20.0 next's chStr is dh left's chStr is null right's chStr is null
T's chStr is odh T's prob is 15 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is g left's chStr is o right's chStr is dh
T's chStr is d T's prob is 5 T's code is 10110 T's numBits is 5
T's entropy is 25.0 next's chStr is h left's chStr is null right's chStr is null
T's chStr is dh T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is i left's chStr is d right's chStr is h

T's chStr is h T's prob is 5 T's code is 10111 T's numBits is 5
 T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
 T's chStr is iodhge T's prob is 60 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is aiodhge left's chStr is iodh right's chStr is ge
 T's chStr is g T's prob is 15 T's code is 110 T's numBits is 3
 T's entropy is 45.0 next's chStr is e left's chStr is null right's chStr is null
 T's chStr is ge T's prob is 35 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is a left's chStr is g right's chStr is e
 T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
 T's entropy is 60.0 next's chStr is iodh left's chStr is null right's chStr is null

*****PostOrder Traversal*****

T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1
 T's entropy is 40.0 next's chStr is iodhge left's chStr is null right's chStr is null
 T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
 T's entropy is 30.0 next's chStr is odh left's chStr is null right's chStr is null
 T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
 T's entropy is 20.0 next's chStr is dh left's chStr is null right's chStr is null
 T's chStr is d T's prob is 5 T's code is 10110 T's numBits is 5
 T's entropy is 25.0 next's chStr is h left's chStr is null right's chStr is null
 T's chStr is h T's prob is 5 T's code is 10111 T's numBits is 5
 T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
 T's chStr is dh T's prob is 10 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is i left's chStr is d right's chStr is h
 T's chStr is odh T's prob is 15 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is g left's chStr is o right's chStr is dh
 T's chStr is iodh T's prob is 25 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is ge left's chStr is i right's chStr is odh
 T's chStr is g T's prob is 15 T's code is 110 T's numBits is 3
 T's entropy is 45.0 next's chStr is e left's chStr is null right's chStr is null
 T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
 T's entropy is 60.0 next's chStr is iodh left's chStr is null right's chStr is null
 T's chStr is ge T's prob is 35 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is a left's chStr is g right's chStr is e
 T's chStr is iodhge T's prob is 60 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is aiodhge left's chStr is iodh right's chStr is ge
 T's chStr is aiodhge T's prob is 100 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iodhge

Output File from Data2:

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is v left's chStr is null right's chStr is null
) ->

(T's chStr is v T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is m left's chStr is null right's chStr is null
) ->

(T's chStr is m T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is c left's chStr is null right's chStr is null
) ->

(T's chStr is c T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null
) ->

(T's chStr is g T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is s left's chStr is null right's chStr is null
) ->

(T's chStr is s T's prob is 2 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is t left's chStr is null right's chStr is null
) ->

(T's chStr is t T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null
) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is b left's chStr is null right's chStr is null
) ->

(T's chStr is b T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null
) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

printing the entropy Table

output T.chStr a T.prob 40 T.code 0 T.numBits 1 T.entropy40.0
output T.chStr i T.prob 10 T.code 100 T.numBits 3 T.entropy30.0
output T.chStr o T.prob 5 T.code 1010 T.numBits 4 T.entropy20.0
output T.chStr h T.prob 1 T.code 101100 T.numBits 6 T.entropy6.0
output T.chStr c T.prob 1 T.code 1011010 T.numBits 7 T.entropy7.0
output T.chStr g T.prob 1 T.code 1011011 T.numBits 7 T.entropy7.0
output T.chStr d T.prob 3 T.code 10111 T.numBits 5 T.entropy15.0
output T.chStr v T.prob 1 T.code 1100000 T.numBits 7 T.entropy7.0
output T.chStr m T.prob 1 T.code 1100001 T.numBits 7 T.entropy7.0
output T.chStr s T.prob 2 T.code 110001 T.numBits 6 T.entropy12.0
output T.chStr t T.prob 5 T.code 11001 T.numBits 5 T.entropy25.0
output T.chStr b T.prob 10 T.code 1101 T.numBits 4 T.entropy40.0
output T.chStr e T.prob 20 T.code 111 T.numBits 3 T.entropy60.0

The Huffman Coding scheme's entropy = 276

*****PreOrder Traversal*****

T's chStr is aiohcgdvmtbe T's prob is 100 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iohcgdvmtbe
T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1
T's entropy is 40.0 next's chStr is iohcgdvmtbe left's chStr is null right's chStr is null
T's chStr is iohcgdvmtbe T's prob is 60 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is aiohcgdvmtbe left's chStr is iohcgd right's chStr is vmstbe
T's chStr is iohcgd T's prob is 21 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is vmstbe left's chStr is i right's chStr is ohcgd
T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
T's entropy is 30.0 next's chStr is ohcgd left's chStr is null right's chStr is null
T's chStr is ohcgd T's prob is 11 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is vmstb left's chStr is o right's chStr is hcgd
T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
T's entropy is 20.0 next's chStr is hcgd left's chStr is null right's chStr is null
T's chStr is hcgd T's prob is 6 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is vmst left's chStr is hcg right's chStr is d
 T's chStr is hcg T's prob is 3 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is d left's chStr is h right's chStr is cg
 T's chStr is h T's prob is 1 T's code is 101100 T's numBits is 6
 T's entropy is 6.0 next's chStr is cg left's chStr is null right's chStr is null
 T's chStr is cg T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vm left's chStr is c right's chStr is g
 T's chStr is c T's prob is 1 T's code is 1011010 T's numBits is 7
 T's entropy is 7.0 next's chStr is g left's chStr is null right's chStr is null
 T's chStr is g T's prob is 1 T's code is 1011011 T's numBits is 7
 T's entropy is 7.0 next's chStr is h left's chStr is null right's chStr is null
 T's chStr is d T's prob is 3 T's code is 10111 T's numBits is 5
 T's entropy is 15.0 next's chStr is vms left's chStr is null right's chStr is null
 T's chStr is vmstbe T's prob is 39 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is a left's chStr is vmstb right's chStr is e
 T's chStr is vmstb T's prob is 19 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is e left's chStr is vmst right's chStr is b
 T's chStr is vmst T's prob is 9 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is b left's chStr is vms right's chStr is t
 T's chStr is vms T's prob is 4 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is t left's chStr is vm right's chStr is s
 T's chStr is vm T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is s left's chStr is v right's chStr is m
 T's chStr is v T's prob is 1 T's code is 1100000 T's numBits is 7
 T's entropy is 7.0 next's chStr is m left's chStr is null right's chStr is null
 T's chStr is m T's prob is 1 T's code is 1100001 T's numBits is 7
 T's entropy is 7.0 next's chStr is c left's chStr is null right's chStr is null
 T's chStr is s T's prob is 2 T's code is 110001 T's numBits is 6
 T's entropy is 12.0 next's chStr is hcg left's chStr is null right's chStr is null
 T's chStr is t T's prob is 5 T's code is 11001 T's numBits is 5
 T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
 T's chStr is b T's prob is 10 T's code is 1101 T's numBits is 4
 T's entropy is 40.0 next's chStr is i left's chStr is null right's chStr is null
 T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
 T's entropy is 60.0 next's chStr is iohcgd left's chStr is null right's chStr is null

*****InOrder Traversal*****

T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1
 T's entropy is 40.0 next's chStr is iohcgdvmstbe left's chStr is null right's chStr is null
 T's chStr is aiohcgdvmstbe T's prob is 100 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iohcgdvmstbe
 T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
 T's entropy is 30.0 next's chStr is ohcgd left's chStr is null right's chStr is null
 T's chStr is iohcgd T's prob is 21 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmstbe left's chStr is i right's chStr is ohcgd

T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
 T's entropy is 20.0 next's chStr is hcgd left's chStr is null right's chStr is null
 T's chStr is ohcgd T's prob is 11 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmstb left's chStr is o right's chStr is hcgd
 T's chStr is h T's prob is 1 T's code is 101100 T's numBits is 6
 T's entropy is 6.0 next's chStr is cg left's chStr is null right's chStr is null
 T's chStr is hcg T's prob is 3 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is d left's chStr is h right's chStr is cg
 T's chStr is c T's prob is 1 T's code is 1011010 T's numBits is 7
 T's entropy is 7.0 next's chStr is g left's chStr is null right's chStr is null
 T's chStr is cg T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vm left's chStr is c right's chStr is g
 T's chStr is g T's prob is 1 T's code is 1011011 T's numBits is 7
 T's entropy is 7.0 next's chStr is h left's chStr is null right's chStr is null
 T's chStr is hcgd T's prob is 6 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmst left's chStr is hcg right's chStr is d
 T's chStr is d T's prob is 3 T's code is 10111 T's numBits is 5
 T's entropy is 15.0 next's chStr is vms left's chStr is null right's chStr is null
 T's chStr is iohcgdvmstbe T's prob is 60 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is aiohcgdvmstbe left's chStr is iohcgd right's chStr is vmstbe
 T's chStr is v T's prob is 1 T's code is 1100000 T's numBits is 7
 T's entropy is 7.0 next's chStr is m left's chStr is null right's chStr is null
 T's chStr is vm T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is s left's chStr is v right's chStr is m
 T's chStr is m T's prob is 1 T's code is 1100001 T's numBits is 7
 T's entropy is 7.0 next's chStr is c left's chStr is null right's chStr is null
 T's chStr is vms T's prob is 4 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is t left's chStr is vm right's chStr is s
 T's chStr is s T's prob is 2 T's code is 110001 T's numBits is 6
 T's entropy is 12.0 next's chStr is hcg left's chStr is null right's chStr is null
 T's chStr is vmst T's prob is 9 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is b left's chStr is vms right's chStr is t
 T's chStr is t T's prob is 5 T's code is 11001 T's numBits is 5
 T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
 T's chStr is vmstb T's prob is 19 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is e left's chStr is vmst right's chStr is b
 T's chStr is b T's prob is 10 T's code is 1101 T's numBits is 4
 T's entropy is 40.0 next's chStr is i left's chStr is null right's chStr is null
 T's chStr is vmstbe T's prob is 39 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is a left's chStr is vmstb right's chStr is e
 T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
 T's entropy is 60.0 next's chStr is iohcgd left's chStr is null right's chStr is null
 *****PostOrder Traversal*****
 T's chStr is a T's prob is 40 T's code is 0 T's numBits is 1

T's entropy is 40.0 next's chStr is iohcgdvmstbe left's chStr is null right's chStr is null
 T's chStr is i T's prob is 10 T's code is 100 T's numBits is 3
 T's entropy is 30.0 next's chStr is ohcgd left's chStr is null right's chStr is null
 T's chStr is o T's prob is 5 T's code is 1010 T's numBits is 4
 T's entropy is 20.0 next's chStr is hcgd left's chStr is null right's chStr is null
 T's chStr is h T's prob is 1 T's code is 101100 T's numBits is 6
 T's entropy is 6.0 next's chStr is cg left's chStr is null right's chStr is null
 T's chStr is c T's prob is 1 T's code is 1011010 T's numBits is 7
 T's entropy is 7.0 next's chStr is g left's chStr is null right's chStr is null
 T's chStr is g T's prob is 1 T's code is 1011011 T's numBits is 7
 T's entropy is 7.0 next's chStr is h left's chStr is null right's chStr is null
 T's chStr is cg T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vm left's chStr is c right's chStr is g
 T's chStr is hcg T's prob is 3 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is d left's chStr is h right's chStr is cg
 T's chStr is d T's prob is 3 T's code is 10111 T's numBits is 5
 T's entropy is 15.0 next's chStr is vms left's chStr is null right's chStr is null
 T's chStr is hcgd T's prob is 6 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmst left's chStr is hcg right's chStr is d
 T's chStr is ohcgd T's prob is 11 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmstb left's chStr is o right's chStr is hcgd
 T's chStr is iohcgd T's prob is 21 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is vmstbe left's chStr is i right's chStr is ohcgd
 T's chStr is v T's prob is 1 T's code is 1100000 T's numBits is 7
 T's entropy is 7.0 next's chStr is m left's chStr is null right's chStr is null
 T's chStr is m T's prob is 1 T's code is 1100001 T's numBits is 7
 T's entropy is 7.0 next's chStr is c left's chStr is null right's chStr is null
 T's chStr is vm T's prob is 2 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is s left's chStr is v right's chStr is m
 T's chStr is s T's prob is 2 T's code is 110001 T's numBits is 6
 T's entropy is 12.0 next's chStr is hcg left's chStr is null right's chStr is null
 T's chStr is vms T's prob is 4 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is t left's chStr is vm right's chStr is s
 T's chStr is t T's prob is 5 T's code is 11001 T's numBits is 5
 T's entropy is 25.0 next's chStr is o left's chStr is null right's chStr is null
 T's chStr is vmst T's prob is 9 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is b left's chStr is vms right's chStr is t
 T's chStr is b T's prob is 10 T's code is 1101 T's numBits is 4
 T's entropy is 40.0 next's chStr is i left's chStr is null right's chStr is null
 T's chStr is vmstb T's prob is 19 T's code is T's numBits is 0
 T's entropy is 0.0 next's chStr is e left's chStr is vmst right's chStr is b
 T's chStr is e T's prob is 20 T's code is 111 T's numBits is 3
 T's entropy is 60.0 next's chStr is iohcgd left's chStr is null right's chStr is null
 T's chStr is vmstbe T's prob is 39 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is vmstb right's chStr is e
T's chStr is iohcgdvmstbe T's prob is 60 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is aiohcgdvmstbe left's chStr is iohcgd right's chStr is vmstbe
T's chStr is aiohcgdvmstbe T's prob is 100 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is a right's chStr is iohcgdvmstbe

deBugFile from Data 1:

Entering constructHuffmanLList method

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is a

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is o

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null

) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is o

newNode.data is e

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null

) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null

) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is h

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null
) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

In listInsert method

Returns from findSpot where Spot.data is o

newNode.data is i

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null
) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null
) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

In listInsert method

Returns from findSpot where Spot.data is i

newNode.data is g

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null

) ->

(T's chStr is h T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null

) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null

) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null

) ->

(T's chStr is g T's prob is 15 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null

) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is d

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null

) ->

(T's chStr is d T's prob is 5 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null

debugFile from Data 1 contains more than 3 pages

deBugFile from Data2:

Entering constructHuffmanLList method

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is a

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is d

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null

) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is d

newNode.data is e

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null

) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null

) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is h

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

In listInsert method

Returns from findSpot where Spot.data is d

newNode.data is i

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null
) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

In listInsert method

Returns from findSpot where Spot.data is dummy

newNode.data is g

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null

) ->

(T's chStr is g T's prob is 1 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null

) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null

) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null

) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null

) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null

) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null

) ->

In listInsert method

Returns from findSpot where Spot.data is d

newNode.data is o

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null

) ->

(T's chStr is g T's prob is 1 T's code is T's numBits is 0

T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null

) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

(T's chStr is d T's prob is 3 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is o left's chStr is null right's chStr is null
) ->

(T's chStr is o T's prob is 5 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is i left's chStr is null right's chStr is null
) ->

(T's chStr is i T's prob is 10 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is e left's chStr is null right's chStr is null
) ->

(T's chStr is e T's prob is 20 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is a left's chStr is null right's chStr is null
) ->

(T's chStr is a T's prob is 40 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is null left's chStr is null right's chStr is null
) ->

In listInsert method

Returns from findSpot where Spot.data is h

newNode.data is s

listHead->(T's chStr is dummy T's prob is 0 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is g left's chStr is null right's chStr is null
) ->

(T's chStr is g T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is h left's chStr is null right's chStr is null
) ->

(T's chStr is h T's prob is 1 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is s left's chStr is null right's chStr is null
) ->

(T's chStr is s T's prob is 2 T's code is T's numBits is 0
T's entropy is 0.0 next's chStr is d left's chStr is null right's chStr is null
) ->

debugFile data2 contains more than 3 pages