

Algorithm Design And Analysis

Project 6

By Shashank Pallerla

Texas State ID: **A04897382**

Huffman Encoding/Decoding and Greedy Algorithm

Source Code:

```
import java.util.PriorityQueue;
import java.util.Scanner;

class HuffmanCoding {
    public static int x = 0;
    public static StringBuilder save;

    public static int varLength = 0;
    public static int totalfixedbytes = 0;
    public static int fixedlength = 0;

    static StringBuffer value = new StringBuffer("");
    private static String binaryformss(int num) {
        StringBuilder buf1 = new StringBuilder();
        StringBuilder buf2 = new StringBuilder();
        while (num != 0) {
            int digit = num % 2;
            buf1.append(digit);
            num = num / 2;
        }
        String binary = buf1.reverse().toString();
        int length = binary.length();
        if (length < fixedlength) {
            while (fixedlength - length > 0) {
                buf2.append("0");
                length++;
            }
        }
        String bin = buf2.toString() + binary;
        return bin;
    }
    private static StringBuffer binaryform(int num) {
        int remainder;
        if (num <= 1) {
            String n = String.valueOf(num);
            value.append(n);
        } else {
            remainder = num % 2;
            String m = String.valueOf(remainder);
            value.append(m);
            binaryform(num >> 1);
        }
        return value;
    }
    public static HTree build(int[] c) {
        PriorityQueue < HTree > trees = new PriorityQueue < HTree > ();
        for (int i = 0; i < c.length; i++)
            if (c[i] > 0)
```

```

        trees.offer(new HuffmanLeaf(c[i], (char) i));
    while (trees.size() > 1) {
        HTree a = trees.poll();
        HTree b = trees.poll();
        trees.offer(new HNode(a, b));
    }
    return trees.poll();
}

public static void print(HTree tree, StringBuffer s) {
    if (tree instanceof HuffmanLeaf) {
        HuffmanLeaf hl = (HuffmanLeaf) tree;
        varLength += hl.frequency * s.length();
        totalfixedbytes += hl.frequency * fixedlength;
        System.out.println(" \t " + hl.value + " \t " +
hl.frequency + "\t \t" + s + "\t \t" +
binaryformss(x));
        x++;
    } else if (tree instanceof HNode) {
        HNode node = (HNode) tree;
        s.append('0');
        print(node.left, s);
        s.deleteCharAt(s.length() - 1);
        s.append('1');
        print(node.right, s);
        s.deleteCharAt(s.length() - 1);
    }
}

public static void main(String[] args) {
    String test;
    for (int i = 0; i < 3; i++) {
        System.out.println("\nPlease enter a text of 2-3 lines:\n");
        Scanner in = new Scanner(System.in);
        test = in.nextLine();
        System.out.println("\n");
        int[] charf = new int[125];
        for (char c: test.toCharArray())
            charf[c]++;
        int count = test.length();
        fixedlength = binaryform(count).length();
        totalfixedbytes = 0;
        HTree t = build(charf);
        System.out.println("\tCharacter \t\t Frequency \t\t Variable Coding
Fixed Coding");

        System.out.println("\t*****
*****");
        print(t, new StringBuffer());
        System.out.println("\n\nComparision of Fixed Length and Variable
Length \n\n ");
        System.out.println("*****");
        System.out.println("\nInput Text : " + test + "\n");
        System.out.println("*****");
    }
}

```

```

        System.out.println("\nText after Decompressing using variable
length coding is : " + test + "\n");
        System.out.println("*****");
        System.out.println("\n Text after Decompressing using fixed
length coding is : " + test + "\n");
        System.out.println("*****");
        System.out.println("\n Character Count:\n Variable Length: " +
varLength + " Characters");
        System.out.println("\n Fixed Length: " + totalfixedbytes + "
Characters");
        System.out.println("\n Characters Saved: " + (totalfixedbytes -
varLength) + " Characters");
        System.out.println("\n*****");

    }
}
}
abstract class HTree implements Comparable < HTree > {
    public final int frequency;
    public HTree(int freq) {
        frequency = freq;
    }
    public int compareTo(HTree tree) {
        return frequency - tree.frequency;
    }
}
class HuffmanLeaf extends HTree {
    public final char value;
    public HuffmanLeaf(int freq, char val) {
        super(freq);
        value = val;
    }
}

class HNode extends HTree {
    public final HTree left, right;

    public HNode(HTree l, HTree r) {
        super(l.frequency + r.frequency);
        left = l;

        right = r;
    }
}

```

Output:

Please enter a text of 2-3 lines:

The quick brown fox jumps over a lazy dog. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character	Frequency	Variable Coding	Fixed Coding

e	9	000	0000000
s	5	0010	0000001
o	5	0011	0000010
l	5	0100	0000011
f	1	010100	0000100
q	1	010101	0000101
.	3	01011	0000110
u	6	0110	0000111
r	6	0111	0001000
d	3	10000	0001001
g	2	100010	0001010
v	2	100011	0001011
i	7	1001	0001100
	16	101	0001101
x	1	1100000	0001110
V	1	1100001	0001111
c	2	110001	0010000
p	4	11001	0010001
n	2	110100	0010010
T	1	1101010	0010011
k	1	1101011	0010100
I	1	1101100	0010101
z	1	1101101	0010110
h	1	1101110	0010111
j	1	1101111	0011000
a	4	11100	0011001
t	4	11101	0011010
b	2	111100	0011011
w	1	1111010	0011100
y	1	1111011	0011101
m	5	11111	0011110

Comparison of Fixed Length and Variable Length

Input Text : The quick brown fox jumps over a lazy dog. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using variable length coding is : The quick brown fox jumps over a lazy dog. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using fixed length coding is : The quick brown fox jumps over a lazy dog. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character Count:
Variable Length: 468 Characters

Fixed Length: 728 Characters

Characters Saved: 260 Characters

Please enter a text of 2-3 lines:

Pack my box with five dozen liquor jugs. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character	Frequency	Variable Coding	Fixed Coding

e	9	000	0000000011111
r	5	0010	00000000100000
t	5	0011	00000000100001
l	5	0100	00000000100010
f	1	010100	00000000100011
q	1	010101	00000000100100
a	3	01011	00000000100101
s	5	0110	00000000100110
.	3	01110	00000000100111
d	3	01111	00000000101000
u	6	1000	00000000101001
p	3	10010	00000000101010
g	2	100110	00000000101011
v	2	100111	00000000101100
	15	101	00000000101101
x	1	1100000	00000000101110
V	1	1100001	00000000101111
c	2	110001	00000000110000
n	2	110010	00000000110001
P	1	1100110	00000000110010
k	1	1100111	00000000110011
I	1	1101000	00000000110100
z	1	1101001	00000000110101
h	1	1101010	00000000110110
j	1	1101011	00000000110111
b	2	110110	00000000111000
w	1	1101110	00000000111001
y	1	1101111	00000000111010
o	4	11100	00000000111011
m	5	11101	00000000111100
i	9	1111	00000000111101

Comparision of Fixed Length and Variable Length

Input Text : Pack my box with five dozen liquor jugs. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using variable length coding is : Pack my box with five dozen liquor jugs. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using fixed length coding is : Pack my box with five dozen liquor jugs. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character Count:
Variable Length: 927 Characters

Fixed Length: 1428 Characters

Characters Saved: 501 Characters

Please enter a text of 2-3 lines:

Jackdaws love my big sphinx of quartz. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character	Frequency	Variable Coding	Fixed Coding

l	5	0000	00000000000000111110
m	5	0001	00000000000000111111
v	2	00100	00000000000001000000
d	3	00101	00000000000001000001
r	5	0011	00000000000001000010
a	5	0100	00000000000001000011
t	5	0101	00000000000001000100
u	5	0110	00000000000001000101
s	6	0111	00000000000001000110
o	3	10000	00000000000001000111
z	1	100010	00000000000001001000
y	1	1000110	00000000000001001001
V	1	1000111	00000000000001001010
.	3	10010	00000000000001001011
h	1	1001100	00000000000001001100
w	1	1001101	00000000000001001101
q	1	1001110	00000000000001001110
f	1	1001111	00000000000001001111
	14	101	00000000000001010000
e	8	1100	00000000000001010001
i	8	1101	00000000000001010010
c	2	111000	00000000000001010011
n	2	111001	00000000000001010100
b	2	111010	00000000000001010101
k	1	1110110	00000000000001010110
x	1	1110111	00000000000001010111
I	1	1111000	00000000000001011000
J	1	1111001	00000000000001011001
g	2	111101	00000000000001011010
p	4	11111	00000000000001011011

Comparision of Fixed Length and Variable Length

Input Text : Jackdaws love my big sphinx of quartz. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using variable length coding is : Jackdaws love my big sphinx of quartz. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Text after Decompressing using fixed length coding is : Jackdaws love my big sphinx of quartz. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.

Character Count:

Variable Length: 1376 Characters

Fixed Length: 2100 Characters

Characters Saved: 724 Characters

Comparison Table with number of characters used:

Text Input	Variable Length Encoding	Fixed Length Encoding	Huffman Coding saved characters
The quick brown fox jumps over a lazy dog. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.	468	728	260
Pack my box with five dozen liquor jugs. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.	927	1428	501
Jackdaws love my big sphinx of quartz. In sagittis semper imperdiet. Vestibulum vel ullamcorper dui.	1376	2100	724