***Problem 1 (Huffman Coding):***

0

0

0

0

1

1

1

1

F

G

H

I

J

**The tree shown below is a Huffman code tree**

**for the letters F, G, H, I, and J. What is the last**

**character in the string encoded by “011101011101"?**

***Solution 1:***

**The code for “F” is “00”**

**The code for “G” is “01”**

**The code for “H” is “10”**

**The code for “I” is “110”**

**The code for “J” is “111”**

**Lets divide the given string into small codes representing each of the letters above we get:**

**“01-110-10-111-01”**

**Replacing them with the relative letters we get:**

**“GIHJG”**

**Therefore the last character in the given string is G.**

***Problem 2 (Huffman Coding):***

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Freq** | **Code** | **Bits** |
| **C** | **32** | **1110** | **4** |
| **D** | **42** | **101** | **3** |
| **E** | **120** | **0** | **1** |
| **K** | **7** | **111101** | **6** |
| **L** | **42** | **110** | **3** |
| **M** | **24** | **11111** | **5** |
| **U** | **37** | **100** | **3** |
| **Z** | **2** | **111100** | **6** |

**Consider the following table:**

1. **Find the Expected cost per letter.**
2. **How many bits do we need to represent these letters?**

***Solution 2:***

1. **The expected cost per letter is the sum of the frequency of each letter multiplied by the number of bits needed to represent it, then the whole answer divided by the total frequency.**

**Expected cost per letter = (32x4 + 42x3+ 120x1+ 7x6+ 42x3+ 24x5+ 37x3+ 2x6)/306**

**= 2.56**

1. **Since we have 8 letters, therefore we will need 3 bits to represent them because log2(8) = 3.**

***Problem 3 (Huffman Coding):***

**What will be the bits per letter for the following Huffman coding**

**Letter Freq**

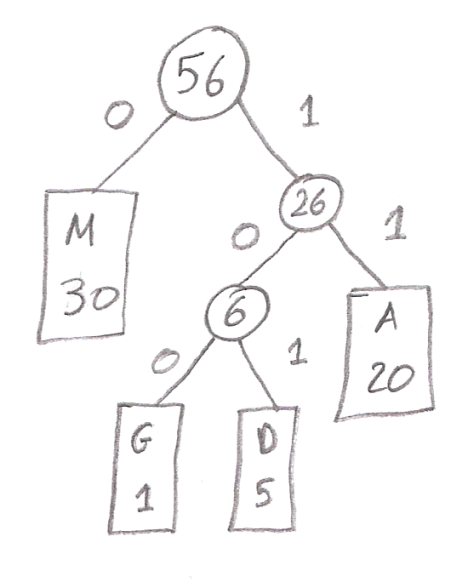
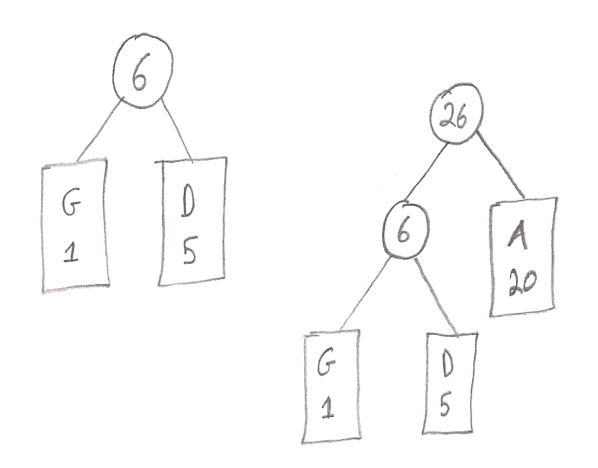
**A 20**

**M 30**

**G 1**

**D 5**

**What will be the encoding of “011101”?**

***Solution 3:***

**The code for “M” is “0”**

**The code for “A” is “11”**

**The code for “G” is “100”**

**The code for “D” is “101”**

**Expected cost per letter = (1x30 + 2x20+ 1x3+ 5x3)/56**

**= 1.57**

**Lets divide the given string into small codes representing each of the letters above we get:**

**“0-11-101”**

**Replacing them with the relative letters we get:**

**“MAD”**

***Problem 4 (Huffman Coding):***

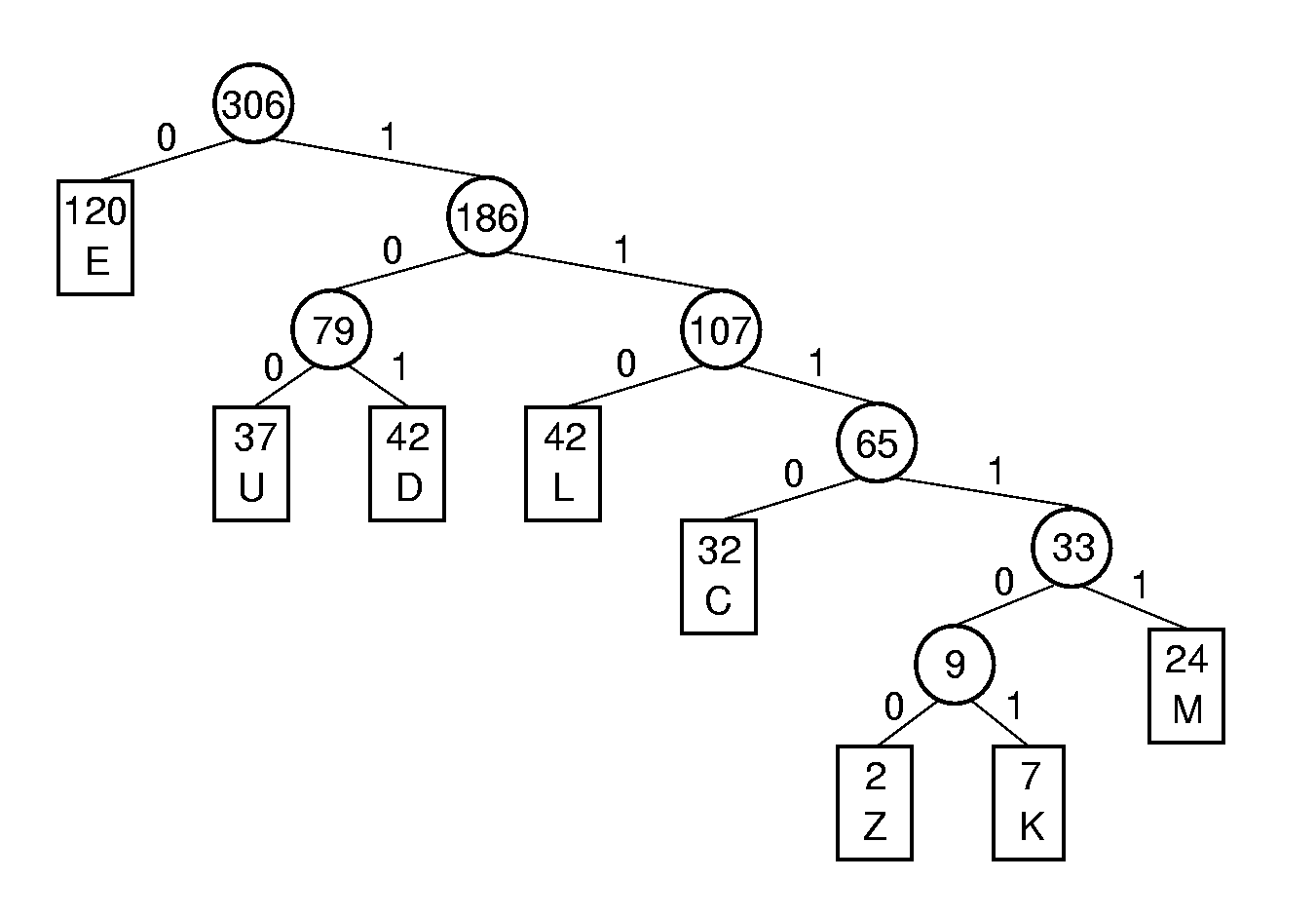
**The variable length coding is:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Z** | **K** | **F** | **C** | **U** | **D** | **L** | **E** |
| **2** | **7** | **24** | **32** | **37** | **42** | **42** | **120** |

**Find the code for the word DEED.**

***Solution 4:***

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Freq** | **Code** | **Bits** |
| **C** | **32** | **1110** | **4** |
| **D** | **42** | **101** | **3** |
| **E** | **120** | **0** | **1** |
| **K** | **7** | **111101** | **6** |
| **L** | **42** | **110** | **3** |
| **M** | **20** | **11111** | **5** |
| **U** | **37** | **100** | **3** |
| **Z** | **2** | **111100** | **6** |

**The Huffman tree is:**

**Code for DEED is: “101 0 0 101”**

***Problem 5 (Huffman Coding):***

**Consider the previous Huffman coding tree. Answer the following questions:**

1. **Decode 1011001110111101.**
2. **Find the Expected cost per letter.**
3. **How many bits do we need to represent these letters?**

***Solution 5:***

1. **Lets divide the given string into small codes representing each of the letters above:**

**“101-100-1110-111101”**

**Replacing them with the relative letters:**

**“DUCK”**

1. **Expected cost per letter = (32x4 + 42x3+ 120x1+ 7x6+ 42x3+ 24x5+ 37x3+ 2x6)/306**

**= 2.56**

1. **Since we have 8 letters, therefore we will need 3 bits to represent them because log2(8) = 3.**

***Problem 6 (Huffman Coding):***

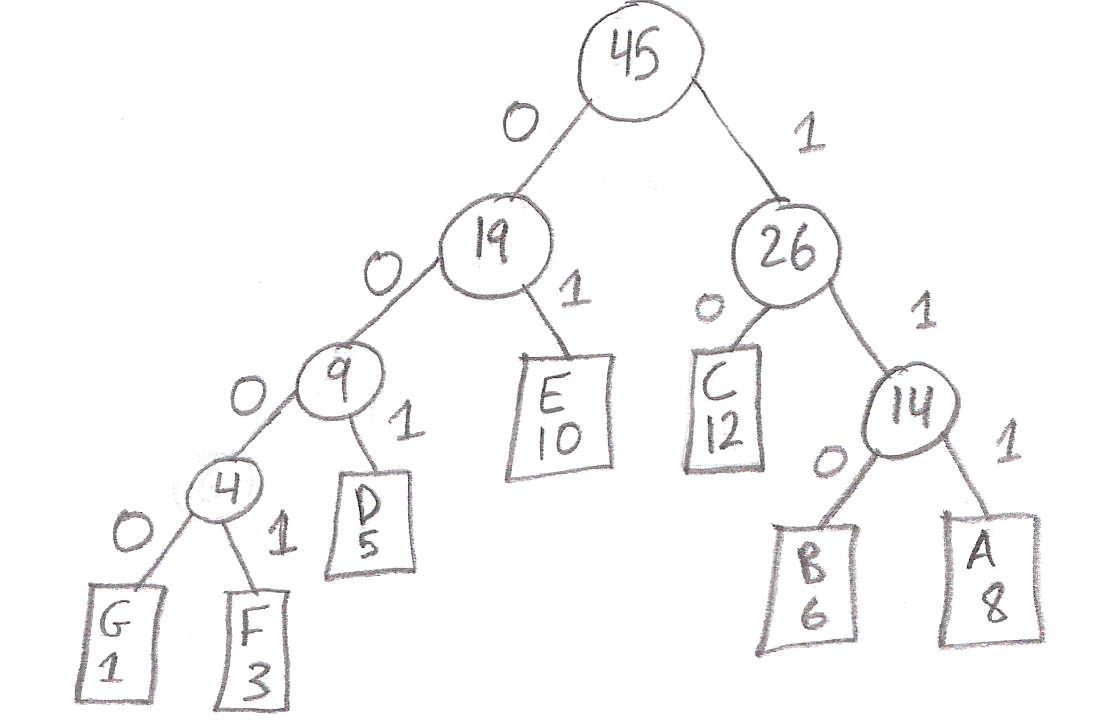
**Given the following table:**

|  |  |
| --- | --- |
| **Symbol** | **Frequency** |
| **A** | **8** |
| **B** | **6** |
| **C** | **12** |
| **D** | **5** |
| **E** | **10** |
| **F** | **3** |
| **G** | **1** |

**What is the code of F?**

***Solution 6:***

**The Huffman Graph will be:**

****

**The code of F will be “0001”.**