Mert Kilic

100334228

Lab 7

CPSC 2150

**Exercise 1**

Diagram

Description automatically generated

|  |  |
| --- | --- |
| Character | Code-Word |
| E | 11 |
| S | 100 |
| N | 101 |
| W | 011 |
| T | 010 |
| Y | 001 |
| I | 000 |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| Character | Code-Word |
| E | 10 |
| S | 11 |
| W | 01 |
| N | 001 |
| T | 0001 |
| I | 00001 |
| Y | 00000 |

1. Using the first trie:

EYEWITNESSNEWS = 11001110110000101011110010010111011100

Using the second trie:

EYEWITNESSNEWS = 10000001001000010001001101111001100111

Both messages are 38 bits long

1. Since Huffman coding is a nondeterministic algorithm, all the possible trees have the same average length. Therefore, having different trees don’t change the compression ratio significantly.

**Exercise 2**

|  |  |
| --- | --- |
| Character | Frequency |
| a | 1 |
| b | 1 |
| c | 3 |
| d | 7 |
| e | 12 |
| f | 18 |
| g | 26 |
| h | 60 |

Map

Description automatically generated

|  |  |
| --- | --- |
| Character | Frequency |
| a | 16 |
| b | 16 |
| c | 16 |
| d | 16 |
| e | 16 |
| f | 16 |
| g | 16 |
| h | 16 |

A diagram of a system

Description automatically generated with low confidence

**Exercise 3**

Well, there are unclear things about Xs algorithm. First, we don’t know that if his algorithm is lossy or lossless. If it is lossy, then it wouldn’t be a very good algorithm since there is a possible data loss. Second, there are better algorithms which reduce the large file size by more than 10%. Huffman coding is one of the examples. However, most of the algorithms do not compress small files very well. Sometimes, small files might get larger as they are compress because of the nature of the compression. Compressing any file by 10% is quite impressive and might be developed to a better algorithm in the future. I think X should be considered because he came up with his own algorithm and looks like it is a greedy algorithm for various cases. World needs X!

**Exercise 4**

1. Time Complexity = dlogn
2. Time Complexity = d