**Lab8- Text Processing**

Họ và tên: Đào Hoài Phương

MSSV: SE172950

**A- Writing exercises**

1. Consider the following code for the letters A through H:
   * 1. 0 **C** 1010 **E** 1100 **G** 1110
     2. 100 **D** 1011 **F** 1101 **H** 1111

What is the result of decoding the string:

100 0 1010 0 1011 0 1100 0 1101 0 100 100 0 0 0 1110 0 1111

**= B-A-C-A-D-A-E-A-F-A-B-B-A-A-A-G-A-H**

1. Given the character frequencies.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **character** | B | C | D | E | F | G |
| **frequency** | 32% | 28% | 16% | 6% | 10% | 8% |

* 1. Using Huffman encoding, what is the code for character F? (Suppose that when constructing a sub tree from 2 nodes we always place node with higher frequency on the left; and the left branch of a node gets value 0, the right one gets value 1).
* **ASC: E-G-F-D-C-B**

BCDFEG

/ \

B CDFEG

/ \

C DFEG

/ \

D FEG

/ \

F EG

/ \

E G

* **F = 100 ||1110**
  1. Encode the text AABCDEFEEFG and calculate the average code length of this string.
  2. Check whether the string 10010101011011001101010111 is a valid encoded string of the above codes or not.

1. What is the maximal length of a codeword possible in a Huffman encoding of an alphabet of n characters?

* **The maximal length of a codeword possible in a Huffman encoding of an alphabet of n characters is n-1**

1. Show that a Huffman tree can be constructed in linear time if the alphabet’s characters are given in a sorted order of their frequencies.

* **A Huffman tree can be constructed in linear time if the alphabet’s characters are given in a sorted order of their frequencies. This is because the algorithm for constructing a Huffman tree involves repeatedly selecting the two nodes with the smallest frequencies and merging them into a new node. If the characters are already sorted by frequency, then this selection process can be done in constant time by simply taking the first two elements of the sorted list. The merging process can also be done in constant time, and since there are n-1 merges to be done for an alphabet of size n, the overall time complexity is O(n).**

1. Given a raw message BBBBBUUUUXXXUUPPPPPPPUUKKKKKKKK.

What is the compressed output if you apply the run-length encoding algorithm for that message?

* **B5U4X3U2P7U2K8**

1. Suppose you are using the LZW algorithm to encode the message.

ABBABABACCDABCCDB and contents of the dictionary at the beginning of encoding are: (1) A (2) B (3) C (4) D. After the encoding process completed:

a. What strings are contained in the dictionary?

**=> The strings contained in the dictionary are: (1) A (2) B (3) C (4) D (5) AB (6) BA (7) BAB (8) AC (9) CD (10) ABC (11) CC (12) DB**

b. What is compressed string?

**=> The compressed string would be: [1,2,6,1,3,5,3,4]**