# CSE221: Algorithms
## Worksheet 7
## Huffman Encoding and Hashmap

1. Using **Variable Length Coding** Scheme encode the following strings.Construct **Huffman Tree**, generate the codeword for each character and show the final encoded message.Also count the total number of bits required to store each string.
   a. Humpty Dumpty had a great fall
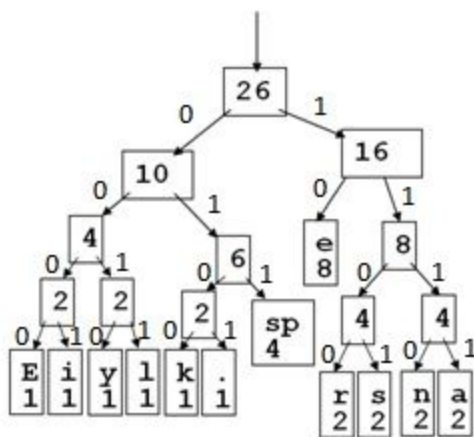   b. Ba Ba Black Sheep

2. Decode the following Encoded Text from the given Huffman Binary Tree.



**Encoded Text:**

0000101100000110011100010101010
1011101101011100111110101111111000
1100111111010010

3. Hash function: KILL = (11+9+12+12) % 11 = 0. Use the Value box to find the values of each character.

**Value:**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

**Hash Table:**

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|
|   |   |   |   |   |   |   |   |   |   |    |

## Operations Table:

| SL NO | Operations | Collision =Yes/No | Hash Function Value | Actual index of the array where the operation is done Or false if invalid/not found | No. of boxes traversed |
|---|---|---|---|---|---|
| 1 | insert(KILL) | No | 0 | 0 | 1 |
| 2 | insert(KK) | Yes | 0 | 1 | 2 |
| 3 | search(EHL) | Yes | 1 | False (not found) | 2 |
| 4 | insert(ZIA) | | | | |
| 5 | insert(AZ) | | | | |
| 6 | delete(KVK) | | | | |
| 7 | insert(KZK) | | | | |
| 8 | insert(AB) | | | | |
| 9 | insert(EXAM) | | | | |
| 10 | delete(KILL) | | | | |
| 11 | insert(LIKE) | | | | |

Execute the operations in sequence listed in the table above and fill-up the columns correctly(in hash table and operations tabel).

    a. For collisions resolution use Linear Probing method.
    b. For collisions resolution use Quadratic Probing method.