CSM-322: Information and Coding Theory
(Odd Semester 2022-23)
Project Part-I

Programming Tasks

Instructions:- 1. Use any programming language in which you feel comfortable.

2. Submit all your programs along with their outputs as a single pdf file with your name, roll no. and branch name.

Q1. **Instantaneous Codes**:
Given an array of alphabets and their corresponding codes. Write a program to check whether the given codes are instantaneous or not.
Example: {A: '1', B: '0'}
Output: Yes
Example: {A: '1', B: '11'}
Output: No

Q2. **Uniquely Decodable Codes**:
Given an input string and a dictionary of words, write a program to find out the number of interpretations of the given string.

Example: '01', {A: '1' , B: '0'}
Output: 1
Explanation: only possible sequence BA
Example: '11', {A: '1', B: '11'}
Output: 2
Explanation: two possible sequences are: AA, B

Q3. **Huffman Codes**:
Given An array of Alphabets and their frequency. Write a program to print Huffman encoding for all the given alphabets.

Example: {A: 12, B: 5}
{A = '1', B = '0'} or {A: '0', B: '1'}

Q4. **Entropy**
Given an array of alphabets and their frequencies. Write a program to

a) calculate the entropy of the corresponding Huffman codes.
b) calculate the entropy for the optimal structure possible for the given frequency array.

Q5. **Hamming Distance:**
Given an array of integers. Write a program to calculate the sum of Hamming distances of all pairs of integers in the array in their binary representation (32 bit).

Example: arr = {1, 3, 5}
Output: 8
Explanation: H(5, 1) + H(5, 3) + H(5, 5) + H(3, 1) + H(3, 3) + H(3, 5) + H(1, 1) + H(1, 3) + H(1, 5) = 8,
where H is Hamming distance.

Q6. **Maximum Likelihood Decoding**:

Suppose that codewords from the code "S" are being sent over a BSC with crossover probability "p". Suppose that the word "**c**" is received. Write a program to find the most likely codeword using maximum likelihood decoding rule.

Example: S = {000, 111}, **c** = 110

if p = 0.05 output is 111

if p = 0.95 output is 000

Q7. **Nearest Neighbour/Minimum Distance Decoding:**

Suppose that codewords from the code "S" are being sent through a noisy channel. Suppose that the word "**c**" is received. Write a program to find the most likely codeword using nearest neighbour decoding.

Example: S = {000, 111}, **c** = 110

Output: 111

Explanation: Hamming_distance(110, 111) < Hamming_distance(110, 000)