Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Data Structures and Algorithms Lab
Assignment 7

Q1. Implement Huffman coding program 'huff'. For encoding, the input to your program is a binary file 'inp.txt' containing a sequence of ASCII bytes. Your program should contain the following:

1. Scan function that scans 'inp.txt' and compute the character/frequency information for each byte present in 'inp.txt'.

2. Huffman coding function that takes as input the character/frequency information and generates Huffman codes.

3. Encode function that takes as input 'inp.txt' and the character/prefixcode table and output the encoding of 'inp.txt' in the specified output file (say 'out.huff'). The output file should first contain the character/prefixcode Huffman table followed by the encoding of 'inp.txt'. The Huffman encoded output should be stored in the output file in the following manner: each block of eight bits of the encoded output is written to the output file as a separate byte. The last byte written in the output file can have padding (if the total bit length of encoded output is not a multiple of eight). The number of padding bits (which is at most seven) can be stored in the beginning of the file immediately after the Huffman table.

4. Stat function that print the following statistics:

   - Size of the input file 'inp.txt' in no. of bytes.
   - Size of the encoded file 'out.huff' in no. of bytes.
   - Percentage compression.
   - Average number of bits per symbol (Total number of bits in output file/ Total number of bytes in input file).

5. Decode function that takes as input a Huffman coded file (output of the encode function) and generates the decoded file. For instance, decoding of 'out.huff' should result in 'inp.txt'.

Usage:

- `huff -e infile outfile` - should encode 'infile' and store the encoding in 'outfile'. The program should also print the statistics.
- `huff -d infile outfile` - should decode 'infile' and store the output in 'outfile'.

Q2. Write a program to generate the following two input files and test the Huffman code program on these two files. Fix $N = 10,000,000$.

File 1: Generate File 1 containing N bytes where each byte is an ASCII code drawn independently and uniformly at random from $\{0, \ldots, 255\}$.

File 2: Generate File 2 containing N bytes where each byte is an ASCII code, say c, drawn independently in the following manner: Draw a number j uniformly at random from $\{0, 1, \ldots, 31\}$. Draw a number m uniformly at random from $\{0, 1, \ldots, 255\}$. Let b denote the position of the most significant bit in m which is set to '1'. If m is 0 then fix $b = 0$. Note that $b \in \{0, 1, \ldots, 7\}$. The next ASCII code c is given by $c = 32b + j$. Note that c takes values from $\{0, \ldots, 255\}$.