

Huffman Codes

Due Mar 30 by 11:59pm

Points 100

Submitting a file upload

Available until May 2 at 11:59pm

This assignment was locked May 2 at 11:59pm.

Write a program that takes any input text and produces both a frequency table and the corresponding Huffman code.

1. Take approximately 360 words from any English document as your input text. Ignore all blanks, all punctuation marks, all special symbols. Create an input file with this input text.
2. Construct the frequency table according to the input text read from the file:
 1. The frequency's must be listed, in order, from largest (at the top) to smallest (at the bottom)
3. Then, using the Huffman algorithm, construct the optimal prefix binary code for the table.
 1. The Huffman codes will be sorted in the same manner as the one above i.e. frequency, highest to lowest.
4. Design your program to read the input from the input file **"infile.dat"**. Your program must produce the output, in the file **"outfile.dat"**, (Files must be named "infile.dat" and "outfile.dat" consisting of
 1. the frequency table for the source text,
 2. the Huffman code for each letter and digit in the source code, and
 3. the length of the coded message in terms of number of bits,

Final Output (Example Values Not Accurate)

Symbol frequency	
A,	15%
m,	11%
7,	6%

Symbol	Huffman Codes
A,	10101
m,	1101
7,	111

Total Bits: 16005

Note: You must not submit your "node_modules" folder if you are working on NodeJs/JavaScript. (Just submit your JavaScript source code and package.json file)

Huffman Coding

Criteria	Ratings		Pts
Prompt user for input file path (if no input from user, assume path: current working directory, file-name: infile.dat)	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Prompt user for output file path. Purge file if exists already, or delete existing contents before writing. (if no input from user, assume path: current working directory, file-name: outfile.dat)	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Read text from input file. (Ignore blanks, punctuation and symbols while parsing)	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Generate frequency table for every symbol in input text. Write frequency table to output file (in DESC order of frequency) (-5 points for round off values).	15.0 pts Full Marks	0.0 pts No Marks	15.0 pts
Generate Huffman codes for every symbol in input text and write to output file (in DESC order of frequency). If frequency table is incorrect, points will be lost on this criterion.	20.0 pts Full Marks	0.0 pts No Marks	20.0 pts
Encoding follows Huffman logic: Most frequent symbol has the smallest binary code. If frequency table is incorrect, points will be lost on this criterion.	20.0 pts Full Marks	0.0 pts No Marks	20.0 pts
Compute length of coded message in terms of Huffman codes. Write this to output file. If frequency table and/or Huffman codes are incorrect, points will be lost on this criterion -5 points, incorrect bits calculation).	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Code style (selection of apt data structures, code efficiency, error checking, etc.)	10.0 pts Full Marks	0.0 pts No Marks	10.0 pts
Output file formatted for easy reading: Tables displayed as matrix, spacing between distinct items (frequency-table, code-table, length of encoded text), etc.	5.0 pts Full Marks	0.0 pts No Marks	5.0 pts
Late submission penalty per policy	0.0 pts Full Marks	0.0 pts No Marks	0.0 pts
			Total Points: 100.0