CSCI 230 -- PA 8

Pattern Matching & Huffman Coding Trees

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). **It is highly recommended that you work with a partner for this PA.** Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Exercise 1: Try Brute Force pattern matching (pseudocode given in C++ textbook), BM pattern matching (code is given), and KMP pattern matching (code given) on various T and P and then modify the code to count the number of comparisons. Use a driver to check them out. Make sure to understand the algorithms.

Try the following test cases:

1. T: “a pattern matching algorithm”, P: “rithm”, P: “rithn”
2. T: “GTTTATGTAGCTTACCTCCTCAAAGCAATACACTGAAAA”, P: “CTGA”, and P: “CTGG”

Exercise 2: Implement a compression scheme that is based on Huffman coding. Write a program that allows user to compress a text file. For example, the two files, ***moneyIn.txt*** and ***moneyOut.txt***, are the normal text file and compressed file respectively (notice the compressed file is in text format so we can easily see what is going on). Given a normal input text file, you need to generate the compressed text file. You should utilize a class named ***HuffmanCoding*** with an appropriate interface. You can use binary tree and PQ data structures from CSCI 220.

File ***moneyIn.txt*** (enter key after the text and then one line of text with no enter key at the end).

```
more money needed
```

File ***moneyOut.txt*** (‘\n’ represents enter key on the first line, first character is a space character on the second line).

```
\n 0110
  1011
```

```
d 100
e 11
m 001
n 000
o 010
r 0111
y 1010
*****
Number of characters: 18
Number of bits: 54
011000101001111110110010100001110101011000111110011100
```

Question 1: Are there any situations where BM better than KMP? Explain.

Question 2: Is there more than one unique Huffman coding tree for a given text? Explain why or why not.

Extra Credit: Modify exercise 1 to accept text (T) from an input data file and pattern (P) from the keyboard. Try the test cases below. Do results seem reasonable? Explain.

1. T: US declaration file from a previous PA, P: "computer", P: "magnanimity"
2. T: humanDNA.txt file on Canvas, P: "`CAAATGGCCTG`", and P: "`CAAATGGGCCTG`"

Fill out and turn in the PA submission file for this assignment (save as PDF format).