**ACS: Problem Set 3**

Due Sunday, July 26 at 11:59PM

**Problem 1: Huffman Encoding**

Suppose that we have the following message:

HEX99CC99HEX99BD99HEX99BD99HEX99FA99

Write a program which will output the following items in this order:

1) The length of the message assuming fixed length identifiers

2) A mapping of the characters to their variable length Huffman encoding identifiers (as in, let us see that “H” → 001, “E” → 0, etc...)

3) The original message (this is for our convenience ^.^)

4) The Huffman encoded message

**Problem 2: All Sources Shortest Paths**

Consider the following adjacency matrix of 12 nodes:

[[0, 55, -1, 87, 52, 22, 17, 15, 15, -1, 18, -1],

[55, 0, 56, -1, 96, 6, -1, 15, 5, 14, 12, -1],

[-1, 56, 0, 88, 15, 13, 48, 67, -1, 41, -1, 9],

[87, -1, 88, 0, -1, -1, -1, -1, -1, 15, -1, 14],

[52, 96, 15, -1, 0, 10, 61, -1, 15, -1, 79, 85],

[22, 6, 13, -1, 10, 0, -1, 70, 13, 5, -1, -1],

[17, -1, 48, -1, 61, -1, 0, -1, -1, 12, 6, 20],

[15, 15, 67, -1, -1, 70, -1, 0, 14, -1, 14, 12],

[15, 5, -1, -1, 15, 13, -1, 14, 0, -1, 85, -1],

[-1, 14, 41, 15, -1, 5, 12, -1, -1, 0, 84, 64],

[18, 12, -1, -1, 79, -1, 6, 14, 85, 84, 0, 56],

[-1, -1, 9, 14, 85, -1, 20, 12, -1, 64, 56, 0]]

In this matrix, the value at row *x* and column *y* means that there exists a direct link from the node numbered *x* to the node numbered *y* with the cost (distance) specified by that value.

Given this information, write a program which prints out a matrix where the value at each position *x*, *y* is the distance of the shortest possible path (of any length) between the point numbered *x* and the point numbered *y*.

Note that we're using -1 to indicate that there is no direct link between the two nodes in question.

**Problem 3: Minimum Spanning Tree**

Consider the adjacency matrix below, where the value at *x, y* is the length from the node numbered *x* to the node numbered *y*. Write a program which returns the weight of the minimum spanning tree for a given adjacency matrix.

[[ 0, 23, 16, 28, 23, 19, 40, 18],

[23, 0, 29, 32, 48, 50, 43, 34],

[16, 29, 0, 25, 41, 27, 23, 12],

[28, 32, 25, 0, 28, 27, 40, 37],

[23, 48, 41, 28, 0, 42, 40, 15],

[19, 50, 27, 27, 42, 0, 27, 49],

[40, 43, 23, 40, 40, 27, 0, 20],

[18, 34, 12, 37, 15, 49, 20, 0]]

**Problem 4: Decryption**

**PLEASE NOTE THAT THIS PROBLEM IS NOT REQUIRED FOR SUBMISSION.**

We're giving you this problem because we think it might be a cool challenge, but we don't want to stress you out with it.

We've attached a file, enc.txt, which has a huge amount of encrypted text written in very typical, modern English. The text was encrypted in a common way: each letter of the alphabets gets converted into another letter. For example, perhaps each "a" gets turned into a "q," each "x" gets turned into a "b," and so on.

Using any method you want, write a program that *helps you* decrypt the text. Note that your program doesn't have to solve it "all in one go." Perhaps your program can print out something *close* to the true text, and then you can edit the decryption mapping a little bit until things look 100% accurate.

Please **don't** just go on the internet to find a one-to-one cipher breaking program. We know that they exist. The goal is for you guys to figure out how to write one.

You should probably use A\*. Ask for help if you get stuck. Be a little bit patient sometimes. If you're using randomness at all, try stuff a few times before you give up. And before you give up, ask around for help some more.

If you do submit, please submit…

1. The first 500 and the last 500 characters of the decrypted text (don't worry about spacing the words properly),
2. Any code that you used to help you decrypt the text, and
3. A description (could be in comments or a separate file) of how you went about solving the problem and who helped you (including "JohnQWhoever on StackOverflow").

Have fun/good luck!