Lab 06

Deadline: 16/04/2022(11:59PM)

IIT Goa

Course Title: **Algorithm Design** Course No: **CS 222**

In this assignment, we have to simulate the communication between two parties in a network. There are two entities, a sender and a receiver. Correspondingly, we have to Implement two modules, *encode.c* and *decode.c*. Further, you are being provided with a source file *message.txt* which contains the information to be communicated. Your program *encode.c* should read the file *message.txt* and generate two files *encoded_data.txt* and $huffmanTable.txt$. As the name indicates, the file *encoded_data.txt* contains 0s and 1s, the binary equivalent of the message, and $huffmanTable.txt$ contains the codewords corresponding to all those characters in the file. The file $huffmanTable.txt$ contains the number of distinct characters in the input file in the first line followed by each line, having a character followed by it's codeword in binary. You may use your heap implementation in your prefix tree construction. The module *decode.c* is easy to implement. This is supposed to read from both the files $huffmanTable.txt$ and *encoded_data.txt* and generate the decoded message and write it to a file *target_message.txt*. You may double check and confirm that the file content is infact what it is intended to be.

OR

In this experiment, the objective is to implement a maplike tool as in the last one using Floyd-warshall algorithm. Your algorithm should preprocess the graph provided in the form of an adjacency matrix in a file *graph.txt* and generate and keep the necessary metadata required to determine the shortest path between any given pair of vertices in $\mathcal{O}(n)$ time where n is the number of vertices.
The file *graph.txt* has the number n in the first line, followed by names of n cities. The third line onwards, the weighted adjascency matrix, where weights are allowed to take negative values (but no negative weighted cycles).

Once the graph being processed, your program should keep on prompting user for source-destination pairs and provide them with the shortest path information, until the program being forcefully terminated. If the input is, say SD, your output should be of the form:

The shortest path is: $S->\ldots->D$ of total cost c

**Recommended reference is textbook (Cormen) but nothing else.

**********END**********