Algorithm Design and Analysis
CSE222 Winter '20

Assignment 3

Due on : 06.03.2020 (9am - 10.15am) (Written)
07.03.2020 (11.59pm) (Programming) Full Marks : 65

Please write solutions independent of each other. You are welcome to discuss with fellow students, but strictly prohibited from copying solutions from another person. No external sources other than class notes/lecture notes uploaded may be consulted for solving these problems. Programming assignment can be done in a team of **at the most two persons**. *Violation of any of the above would be considered an act of plagiarism. Please see collaboration policy on course webpage for further information.*

Note: For each of the theory questions, you *must* give proof of correctness of your algorithm. Without that, you would be awarded no credit *even if your proposed algorithm is correct.*

For programming assignment, you need to submit on Google Classroom by the assigned deadline.

Problem 1 (15 points) Suppose you are given n tasks $J = \{j_1, j_2, \cdots j_n\}$. Each task j_ℓ has two parts - a preprocessing phase which takes p_ℓ units and a main phase which takes f_ℓ units of time. There are n machines that can execute the main phases of the jobs *in parallel*. However, the preprocessing phases need to be executed sequentially on a special machine. The completion time of any schedule is the earliest time when *all tasks* have finished execution. Design a greedy algorithm which produces a schedule that minimizes the completion time.

Problem 2 (20 points) You have been given a *complete binary tree* rooted at r. Each edge e has a positive length ℓ_e. The distance between the root r and any leaf is the total length of all the edges along the path from r to the leaf. As you can imagine, the distance between the root r and any two distinct leaves can be quite different. Your task is to increase the length of some of the edges such that in the resulting tree, *distance between the root and all the leaves will be the same.* Note that you can only elongate edges and cannot shrink them. Design a greedy algorithm that achieves this goal such that the *sum of all edge lengths* in the final tree is minimized. (Hint: recursion and greedy)

Problem 3 (30 points) **(programming problem.)**

You can write your code in either C++ or Java. Note that the demo will be on the basis of the code that *you have uploaded by the deadline on Google Classroom.*

1) Given a text file input.txt, write a program that encodes the file using Huffman's Coding to a Bit-String. Store the encoded string in a file named encoded.txt. Also, store the coding map into a file "mapping.txt".

2) Given some encoded.txt, write a program that decodes it to the plaintext (original text), using the coding map dumped in part (1).

3) The Fano-Shannon coding also produces prefix free codes using a top-down approach. The idea is, split the set of alphabets in to two halves such that cumulative frequencies of the

halves are approximately equal, and then solve recursively. Implement such a coding and compare and report the difference between average access time of this one and the one obtained on the same input through Huffman's code. You are free to choose your own style of output for this part, as long as the answer is correct.

Write programs in separate files for the first 3 questions. Place everything is a zip/tarball file that has to be named as <team member 1 roll no>_<team member 2 roll no.>_pa1.zip

For the demo, you would be given a few input files with names "input.txt". Each of these will consist of an entire string which has to be encoded. These strings may be made up of English alphabets, punctuation marks(.,!?-) or spaces (Suppose no linebreaks). All of the above should be encoded and the output should be stored in an output file "encoded.txt". Please ensure your code doesn't output any meaningless statements and only provides the required output.