

Problem Set 7—Due Tues, March 10 3:10PM

(25) Problem 1. Use network flows to find an efficient solution for the following problem:

a) In a computer network there are p processors P_1, P_2, \dots, P_p , and c communication lines C_1, C_2, \dots, C_c (with $c > p$). Each processor i has the ability to test t_i lines per day and there is a list L_i which contains the communication lines that processor i is able to test. Subject to these constraints we would like to be able to test all the communication lines every day. A testing schedule determines for each processor the lines it should test. Give an efficient algorithm to find a testing schedule or determine that no schedule can test all lines in a single day. Give the run time of your algorithm in terms of p and c .

b) Same as a) but find the minimum number of days to test all lines (and a schedule that achieves it).

(25) Problem 2.

Suppose we are given a directed graph G ;

a) We want to find 3 edge disjoint paths from a designated starting vertex u to a specific destination vertex d (or determine that no such paths exist). Describe an efficient algorithm to find the paths and give the running time of your solution.

b) Now suppose we have three different destinations d_1, d_2 and d_3 . We want to find 3 paths from our start vertex u , one to each of the destinations and we want these 3 paths to be edge disjoint. Describe an efficient algorithm to find the paths and give the running time of your solution.

(25) Problem 3 For the input string: BABAACAABAACACAABAACCBABAAC

(a) Give a Huffman code for this string (describe both your tree and the actual binary encoding of the above string).

(b) Give the "high-level" Lempel-Ziv encoding of the above string where each match of length 3 or longer is represented by a length, pointer pair, and otherwise by the actual input character.

(c) Now give the actual bit string representation for the encoding of part b) that would be produced by encoding literals, lengths and pointers using two Huffman trees. Again give the Huffman trees used followed by the encoding.