Homework Assignment 1

TTIC 31010/CMSC 37000-1

due on January 30, 11:59pm

Please typeset your solutions. Do not discuss the problems with other students. Do not look up solutions online. Submit your work through canvas.

Problem 1. We are given a set of n points $X = \{x_1, \dots, x_n\}$ on the real line. Give an algorithm that finds a minimum cardinality set of unit intervals that cover all points in X. Prove its correctness. Find its running time.

Example. If $X = \{0.1, 0.5, 0.7, 1.6, 2.1\}$, then one optimal solution is $\{[0, 1], [1.5, 2.5]\}$.

Problem 2. We are given $2n$ equally-spaced points on a line. Half the points are black, and half are white. The goal is to connect every black point to a distinct white point with a wire so as to minimize the total wire length. Below are two suggestions for greedy algorithms, which may or may not solve the problem correctly. For each suggested algorithm, if it is correct, prove its correctness. If it is incorrect, prove that by showing an input on which the algorithm does not find an optimal solution. Each of the suggested algorithms performs n iterations. In every iteration, it selects a single pair of points to connect and then deletes these two points from the line. In order to define each algorithm, it is now enough to specify the greedy rule for selecting the pair of points to connect.

1. Greedy Rule 1: find any pair p, p' of points, where p is white and p' is black, and no other point lies between the two. Connect p to p' and delete both points from the line.

2. Greedy Rule 2: Let p be the leftmost black point and let p' be the leftmost white point. Connect p to p' and delete both points from the line.

Problem 3. We are given an alphabet $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ and frequencies p_i of all characters in Σ.

1. Assume that the frequency of some character is strictly greater than $2/5$ (that is, $p_i > 2/5$ for some i). Prove that there is a codeword of length 1 in the Huffman code. (For a partial credit, solve the question assuming that the frequency of some character is strictly greater than $1/2$).

2. Assume that the frequency of every character is strictly less than $1/3$ (that is, $p_i < 1/3$ for every i). Prove that there is no codeword of length 1 in the Huffman code.

Problem 4. Given an unlimited supply of coins of denominations $x_1, x_2, \ldots, x_n$ (where $x_1, \ldots, x_n$ are positive integer numbers), we wish to make change for a value v; that is, we wish to find a set of coins whose total value is v (the set may contain several coins of the same denomination). This might not be possible: for instance, if the denominations are 5 and 10 then we can make change for 15 but not for 12. Design a dynamic programming algorithm, with running time $O(nv)$, that does the following.

1. The algorithm determines if there is a set of coins of total value v.
2. If there is such set, the algorithm finds the set with the minimal possible number of coins.

Describe your algorithm in detail. Prove its correctness.

Problem 5. *Programming assignment. See a separate post for implementation and submission requirements.*

We say that a binary string B is *sparse* if there are no three consecutive 1's in B. For instance, strings $1, 110101, 0001011$ are sparse, but $111, 1100011100, 010101110101$ are not. Given a sequence of non-negative weights $w_0, \ldots, w_{n-1}$, define the value of the string $B = b_0 b_1 \ldots b_{n-1}$ as $\sum_{i=0}^{n-1} b_i w_i$.

Design and implement a polynomial-time algorithm that given a sequence $w_1, \ldots, w_n$ finds the value of the most valuable sparse string.