Homework Assignment 1

Due on February 2

TTIC 31010/CMSC 37000-1

January 20, 2016

You can discuss the problems with other students taking the class. However, you must write your solutions yourself. Please, do not look up solutions online.

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.

Problem 2. We are given an alphabet Σ with 2^k characters and a set of frequencies $p(\sigma)$ such that

$$\min_{x \in \Sigma} p(x) > \frac{1}{2} \max_{x \in \Sigma} p(x).$$

How does the Huffman (prefix) tree for Σ look like? What is its cost? Prove your answer.

Problem 3. Given an unlimited supply of coins of denominations $x_1, x_2, \dots, x_n$ (where $x_1, \dots, x_n$ 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 such set with the minimal possible number of coins.

Describe your algorithm is detail. Prove its correctness.

Problem 4. We say that a set $A \subset \{1, \dots, n\}$ is *good* if among every three consecutive numbers $i, i+1, i+2$ (for $1 \le i \le n-2$) either one or two numbers belong to A. For example, the set $\{1, 2, 4, 5\} \subset \{1, \dots, 6\}$ is good but sets $\{4, 5\} \subset \{1, \dots, 6\}$ and $\{2, 3, 4, 6\} \subset \{1, \dots, 6\}$ are not. Design a polynomial-time algorithm that given n and a sequence $w_1, \dots, w_n$ finds a good set $A \subset \{1, \dots, n\}$ that maximizes the sum $\sum_{i \in A} w_i$. Describe your algorithm in detail. Prove the correctness of your algorithm. Find its running time.