Assignment 3

Question 1. [16 MARKS]

Given a list L, a *contiguous* sublist M of L is a sublist of L whose elements occur in immediate succession in L. For instance, $[4, 7, 2]$ is a contiguous sublist of $[0, 4, 7, 2, 4]$ but $[4, 7, 2]$ is not a contiguous sublist of $[0, 4, 7, 1, 2, 4]$.

We consider the problem of computing, for a list of integers L, a contiguous sublist M of L with maximum possible sum.

Algorithm 1 $MaxSublist(L)$

<precondition>: L is a list of integers.
<postcondition>: Return a contiguous sublist of L with maximum possible sum.

Part (1) [5 MARKS]

Using a divide-and-conquer approach, devise a recursive algorithm which meets the requirements of $MaxSublist$.

Part (2) [8 MARKS]

Give a complete proof of correctness for your algorithm. If you use an iterative subprocess, prove the correctness of this also.

Part (3) [3 MARKS]

Analyze the running time of your algorithm.

Question 2. [18 MARKS]

For a point $x \in \mathbb{Q}$ and a closed interval $I = [a, b]$, $a, b \in \mathbb{Q}$, we say that I covers x if $a \leq x \leq b$. Given a set of points $S = \{x_1, \ldots, x_n\}$ and a set of closed intervals $Y = \{I_1, \ldots, I_k\}$ we say that Y covers S if every point x_i in S is covered by some interval I_j in Y.

In the "Interval Point Cover" problem, we are given a set of points S and a set of closed intervals Y. The goal is to produce a minimum-size subset $Y' \subseteq Y$ such that Y' covers S.

Consider the following greedy strategy for the problem.

Algorithm 2 *Cover*(S, Y)

<precondition>:

S is a finite collection of points in $\mathbb{Q}$. Y is finite set of closed intervals which covers S.

<postcondition>:

Return a subset Z of Y such that Z is the smallest subset of Y which covers S.

$L = \{x_1, \dots, x_n\} \leftarrow S$ sorted in nondecreasing order
$Z \leftarrow \emptyset$
$i \leftarrow 0$
while $i < n$ **do**
    if x_{i+1} is not covered by some interval in Z **then**
        $I \leftarrow$ interval $[a, b]$ in Y which maximizes b subject to $[a, b]$ containing x_{i+1}
        Z.append(I)
    $i \leftarrow i + 1$
return Z

Give a complete proof of correctness for *Cover* subject to its precondition and postcondition.

Question 3. [10 MARKS]

The first three parts of this question deals with properties of regular expressions (this is question 4 from section 7.7 of Vassos' textbook). Two regular expressions R and S are equivalent, written $R \equiv S$ if their underlying language is the same i.e. $\mathcal{L}(R) = \mathcal{L}(S)$. Let R, S, and T be arbitrary regular expression. For each assertion, state whether it is true or false and justify your answer.

Part (1) [2 MARKS]

$$\text{If } RS \equiv SR \text{ then } R \equiv S.$$

Part (2) [2 MARKS]

$$\text{If } RS \equiv RT \text{ and } R \not\equiv \emptyset \text{ then } S \equiv T.$$

Part (3) [2 MARKS]

$$(RS + R)^* R \equiv R(SR + R)^*.$$

Part (4) [4 MARKS]

Prove or disprove the following statement: for every regular expression R, there exists a FA M such that $\mathcal{L}(R) = \mathcal{L}(M)$. *Note*: even if you find the proof of this somewhere else, please try to write up the proof in your own words. Just citing the proof is *NOT* enough.

Question 4. [16 MARKS]

In the following, for each language L over the alphabet $\Sigma = \{0, 1\}$ construct a regular expression R and a DFA M such that $\mathcal{L}(R) = \mathcal{L}(M) = L$. Prove the correctness of your DFA.

Part (1) [8 MARKS]

Let $L_1 = \{x \in \{0,1\}^* :$ the first and last charactes of x are the same$\}$. Note: $\epsilon \notin L$ since ϵ does not have a first or last character.

Part (2) [8 MARKS]

Let a *block* be a maximal sequence of identical characters in a finite string. For example, the string 0010101111 can be broken up into blocks: 00, 1, 0, 1, 0, 1111. Let $L_2 = \{x \in \{0,1\}^* :$ x only contains blocks of length at least three$\}$.