Complexity Qual Spring 2018
(provide proofs for all answers)

Problem 1: Regular Languages:
Show that for every regular language L, the languages below are regular. You can use any definition of what a regular language is for the proofs.

Problem 1a: Show the following language is regular:
$cycle(L) = \{w | w = xy \text{ s.t. } yx \in L \text{ for some strings } x, y\}$.

Problem 1b: Show the following language is regular:
$\max(L) = \{w|w \in L, \text{ but for no non-null string } x, wx \in L\}$.

Problem 3: Context Free Languages: :

Problem 2a: Show the following language is not context-free:
$L = \{a^{n^2} | n \geq 0\}$.

Problem 2b: Show the following language is not context-free:
$L = \{w \in \{a, b, c\}$: |w has an equal number of a's, b's and c's }.

Problem 3: NP-Hardness:

Show that the following problem is NP-Complete:
Given an undirected graph G with arbitrary lengths on the edges (these could be positive as well as negative), a length value L (could be positive or negative), and a vertex s, is there a simple cycle that passes through s, such that the ratio of its length to the number of edges in it (mean length) is at most L?
(You can assume the NP-Completeness of any standard problem, like 3SAT, hamiltonian path, vertex cover, or clique.)

Problem 4: Deterministic Polynomial Time or NP-Complete?:

SNAKE PROBLEM: You are given an undirected graph $G = (V, E)$ with a length function l, where $l(e)$ is the length of edge e, and a weight function w, where $w(v)$ is the weight at vertex v. You are also given a starting vertex s for the snake. You must guide the snake through the graph. The snake initially has 0 length. Every time the head of the snake reaches a vertex v for the first time, the snake eats the weight there, and becomes $w(v)$ longer, extending back along the path it took.

For example, if the snake travels from $s = A$ to B to C, and we have $w(A) = 0, w(B) = 10, w(C) = 5$ and $l(AB) = 12, l(BC) = 6$, then the head of the snake is currently at C, and the snake covers all of edge BC and the last 9 of edge AB, for a total length of 15.

Once the head of the snake touches its own body, the game ends. For example, extending the above example, if the snake continues from C to D to B and $w(D) = 0, l(CD) = 1, l(DB) = 1$, then the snake runs into its own body at B, because the snake still covers the last 7 of edge AB.

An instance of the SNAKE problem consists of a graph G with s, w, and l, and a target length T. You are asked whether the snake can take a walk through the graph starting at s to grow to length at least T, before running into itself. Either give an algorithm that solves this problem in deterministic polynomial time, or prove that it is NP-complete.

Problem 5: Define P(k) to be the set of problems with an alphabet of size k that are decidable by a Turing machine in deterministic time that is polynomial in the length of the input.

Problem 5a: Give an example of an NP-hard problem that is in P(1).

Problem 5b: Show that for any constant $k \geq 2, P(k) = P$.
(Hint: Note that $P = P(2)$ by definition. So, you need to show that $P(k) = P(2)$ for all $k \geq 2$.)

Problem 6: Oracles:
Construct an oracle L such that NP^L is not equal to coNP^L where coNP is the class of problems whose complement is in the complexity class NP.