PROBLEM 1. Review questions.

(a) What does it mean to say that a TM runs in time $t\colon \mathbb{N} \to \mathbb{N}$?

(b) What is the time complexity class $\mathrm{TIME}(t(n))$?

(c) What does it mean for two models of TMs (single-tape, multitape, etc) to be polynomially related?

(d) Give a precise definition of the class P.

PROBLEM 2.

(a) Prove that $n^2 + n = O(n^2)$.

(b) Prove that $O(n)+O(n\log(n)) = O(n\log(n))$. (Note: to say $f(n) = O(n)+O(n\log(n))$ means that there exist constants c, c' and an integer n_0 such that $n \geq n_0$ implies $f(n) \leq cn + c'n\log(n)$. There are two directions to prove.)

(c) For which integers $m \geq 0$ is $m! \geq 2^m$? Give a proof.

PROBLEM 3. True or False. No proof required.

(a) $2n = O(n)$	(b) $n^2 = O(n)$	(c) $n^2 = O(n\ \log^2(n))$
(d) $n\ \log(n) = O(n^2)$	(e) $3^n = 2^{O(n)}$	(f) $2^{2^n} = O(2^{2^n})$.

PROBLEM 4. True or False. No proof required.

(a) $n = o(2n)$	(b) $2n = o(n^2)$	(c) $2^n = o(3^n)$
(d) $1 = o(n)$	(e) $n = o(\log(n))$	(f) $1 = o(1/n)$.

PROBLEM 5. Determine, with proof, whether the complexity class P is closed with respect to the following operations:

(a) union

(b) concatenation

(c) complement.

To prove a closure property holds, outline an algorithm as in Sipser's text: give a list of stages and show that each step in each stage uses a polynomial number of steps. (Here, as usual, *polynomial number* means polynomial in the input size.) Number your stages, indicate the run times for each step, and indicate the number of iterations of any loop that appears as in Sipser's text using big-O notation. Compute the total run time.

PROBLEM 6. Let G be an undirected simple graph. (Take "simple" to mean no multiple edges, no loop edges, and a finite number of vertices.) A *triangle* of edges is a triple of vertices u, v, w such that each pair lies on an edge. Define

$$\mathrm{TRIANGLE} = \{\langle G\rangle : G \text{ is a simple graph containing a triangle}\}.$$

Show that TRIANGLE is in P, giving a description of a polytime algorithm as in Sipser's text. That means describing an algorithm with numbered stages giving the run time for each step in a stage. Taking into account the number of times each loop is iterated, show your algorithm decides in polynomial time.

Note the input size is the length n of any reasonable encoding $\langle G \rangle$ of the graph G. In particular, the number of vertices of G is $O(n)$.

Problem 7. (bonus) Consider the language

MODEXP $= \{\langle a, b, c, m \rangle : a, b, c, m$ are positive integers written in binary and $a^b = c \bmod m\}$.

Show that MODEXP $\in P$. Note that the most obvious algorithm does not run in polynomial time. Hint: first try the case where b is a power of 2.