PROBLEM 1. Review questions.

(a) What is a Hamiltonian path from vertex s to vertex t?

(b) What is a NTM decider? (Does it need to halt on all branches?)

(c) What does it mean to say a NTM runs in time $O(t(n))$? (Do all branches need to end in $O(t(n))$ steps? Only some of them? The sum of the times of all branches?)

(d) Let $t\colon \mathbb{N} \to \mathbb{N}$. What is $NTIME(t(n))$? (Is it a set of NTMs? A language?)

(e) What is the class NP? (Again: what, precisely, are the elements of NP?)

PROBLEM 2. 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

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

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 3. An undirected simple graph G is 3-*colorable* if each of its vertices can be assigned one of three colors in such a way no two adjacent vertices have the same color. (Adjacent vertices are vertices that share an edge.) Define the language

$$\text{3COLOR} = \{\langle G \rangle : G \text{ is 3-colorable}\}.$$

(a) Give some examples of graphs that are 3-colorable and some that are not.

(b) How many ways are there to color a graph with m nodes using 3 colors if there is no restriction on the colors of adjacent vertices?

(c) Show that 3COLOR is in NP by giving a numbered algorithm in the style of those given in Sipser's text. Give the run time for each stage and the total run time in big-O notation.

PROBLEM 4. Let n, b be integers with $n \geq 1$ and $b \geq 2$.

(a) Show that the number of digits of n base b is $\lfloor \log_b(n) \rfloor + 1$. Why isn't it the number of digits $\lceil \log_b(n) \rceil$? (Hint: n has r digits base b if and only if n is between which two powers of b? Be careful with the endpoints of that interval.)

(b) Let $a \geq 2$. Give the proof of the fact that $f(n) = O(\log_a(n))$ iff $f(n) = O(\log_b(n))$. (Thus, the number of digits of n in base b is $O(\log(n))$ no matter which base we choose.)

(c) Let $u(n)$ be the length of n written in unary: $n = \overbrace{1 \cdots 1}^{n\text{-times}}$. Show that $u(n) \neq O(\log(n))$.

(d) Consider the operation $d\colon \mathbb{N} \to \mathbb{N}$ defined by $d(n) = \lfloor n/2 \rfloor$. Next, define $h\colon \mathbb{N} \to \mathbb{N}$ by $h(n) = \min\{k : d^{(k)}(n) = 0\}$. Here, $d^{(k)}$ denotes k-fold composition. Is $h(n) = O(\log(n))$? Prove or disprove.

PROBLEM 5. (bonus) Consider the language

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

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.