Readings: Sections 7.1, 7.2, 7.3

Problem 1: Answer each of the following with TRUE or FALSE. You do not need to justify your answers. (Note: when dealing with sets like $O(f(n))$, $\Omega(f(n))$, etc., we use the symbols = and \in interchangeably.)

1. $3 = O(n)$
2. $12n = O(n)$
3. $n^4 = O(n^3 \log^3(n))$
4. $3n \log(n) + 1000n = O(n^2)$
5. $3^n = O(2^n)$
6. $3^n = 2^{O(n)}$
7. $2^{2^n} = O(2^{2^n})$
8. $n^n = O(n!)$
9. $n = o(3n)$
10. $1000n = o(n^3)$
11. $3^n = o(4^n)$
12. $1000 = o(n)$
13. $n = o(\log^2(n))$
14. $\frac{1}{n} = o(1)$
15. $\log_2(n) = \Theta(\log_{10}(n))$
16. $3^n = \Theta(4^n)$
17. $n^3 = \Theta(g^{\log n})$
18. $n^2 = \Omega(n^3)$
19. $\log(n) = \Omega(\log(\log(n)))$
20. $4^n = \Omega(2^{\pi^2})$

Problem 2: (Sipser problem 7.12)
Let

$$MODEXP = \{ \langle a, b, c, p \rangle \mid a, b, c \text{ and } p \text{ are binary integers such that } a^b \equiv c \pmod{p} \}.$$

Show that $MODEXP$ is in $P$. (Note that the first and the most obvious algorithm you would come up with would run in time exponential in the input length. Hint: Try it first when $b$ is a power of 2.)

Problem 3: (Based on Sipser problem 7.14) Prove that $P$ is closed under:

1. The concatenation operation.
2. The star operation.

Problem 4: Prove that $NP$ is closed under:

1. The intersection operation.
2. The concatenation operation.

Problem 5: Prove that the following languages are in $NP$. You may use either the guess-and-check (certificate/verifier) method, or else describe a nondeterministic Turing machine that decides the language in time polynomial in the length of the input.
1. (From Sipser exercise 7.11)

ISO = \{ (G, H) | G and H are undirected graphs and G and H are isomorphic \}

(Two graphs are isomorphic if, by renaming the nodes of one, we get a graph that is identical to the other.)

2. TRIPLE-SAT = \{ (\phi) | \phi is a Boolean formula and \phi has at least three distinct satisfying assignments \}

(Boolean formulas are defined on p. 271 of Sipser’s book.)

3. A crossword puzzle construction problem is specified by a finite set \( W \subseteq \Sigma^* \) of words, and an \( n \times n \) matrix \( A \) whose entries are either 0 or 1 (intuitively, a 0 corresponds to a blank square, and a 1 corresponds to a black square). The goal is to use the words in \( W \) to fill in the blank squares. Formally, suppose \( E \) is the set of all pairs \((i, j)\) such that \( A_{ij} \), the \((i, j)\)th entry of \( A \), is 0. We want to find a mapping \( f : E \to \Sigma \) such that the letters assigned to any maximal horizontal or vertical contiguous sequence of members of \( E \) form, in order, a word of \( W \). If this is possible, we say that \( (W, A) \) is a constructable crossword system.

\( CROSSWORD = \{ (W, A) \mid W \subseteq \Sigma^* \text{ and } A \text{ is an } n \times n \text{ } 0 - 1 \text{ matrix and} \)

\( (W, A) \text{ is a constructable crossword system.} \}

(For instance, the set \( W = \{ a, b, ab, ba, aba \} \) over the alphabet \( \{0, 1\} \) and the matrix \( A \) as in the figure form a constructable crossword system. One of the crosswords so constructed is the matrix \( B \) in the figure.)