CS 332 - Homework 2

Due: Thursday, February 7

You should keep an extra copy all your homeworks if you do not typeset the answers. To turn in HW 2 and all future homework you should use Gradescope. If needed Gradescope will be reviewed next Monday in the lab sections. Just ask Jaiyu. When you register for Gradescope please use your BU ID and also your full name that is on your BU records.

Reading: Chapter 3, section 3.2 and 3.3 of the Sipser textbook, pages 176-187 And chapter 4, section 4.1, pages 194-201

PROBLEMS: Answer the following 6 questions.

1. For each of the following sets L say whether they are languages and briefly say why or why not.
   i. \( L = \{ \text{the binary string } b_0, b_1, b_2, ..., |b_i| = 1 \text{ whenever the } i \text{ is a prime number } \} \)
   ii. \( L = \{ \text{the real number } \pi \} \)
   iii. \( L = \{ 2, 3, 4 \}^* \)

2. Let \( A \) be any fixed finite set of 4 or more elements. Prove that the number of permutations of elements in \( A \) is greater than the number of subsets of elements of \( A \).

3. Consider the statement \( \forall x \in S \exists y \leq x \exists z \leq x \in S(yz = x) \).
   i. Is this statement true when \( S = N^+ \)? \( S = N \)? When \( S = Z \)? When \( S = Q \)?
   ii. Explain your answer for each question using no more than two sentences.
   iii. What statement is the negation of each of the first statement above?
   iv. Answer the 3 questions for the negation.

4. Consider the following 2 step procedure \( P \) and explain why \( P \) is not a legitimate algorithm of the kind we discussed in class.

   \( P: \) The input to \( P \) is three natural numbers \( a, b, c \).
   
   Step 1. Try all possible assignments of natural numbers to \( a, b, c \) and for each of these possible settings test if \( a^3 + b^3 = c^3 \)
   
   Step 2. If the test you make of the equation in 1 is ever true, then \( P \) outputs true, if all of the tests you make in 1 are false then \( P \) outputs false.
5. Answer problem 0.12 on page 27 of our textbook.

6. A k-clique in a graph $G$ is complete subgraph of $G$ with $k$ vertices. (So for example, $G$ has a 3-clique if it contains some triangle.)
   i. Give an example of a graph with a 4-clique but which doesn’t have a 5-clique.
   ii. Describe an algorithm which takes as input a graph $G$ and which decides if $G$ has a 3-clique or not.

   What is the order of the (time) complexity of your algorithm?