Recall fooling sets and distinguishability. Two strings \( x, y \in \Sigma^* \) are suffix distinguishable with respect to a given language \( L \) if there is a string \( z \) such that exactly one of \( xz \) and \( yz \) is in \( L \). This means that any DFA that accepts \( L \) must necessarily take \( x \) and \( y \) to different states from its start state. A set of strings \( F \) is a fooling set for \( L \) if any pair of strings \( x, y \in F, x \neq y \) are distinguishable. This means that any DFA for \( L \) requires at least \( |F| \) states. To prove non-regularity of a language \( L \), you need to find an infinite fooling set \( F \) for \( L \). Given a language \( L \), try to find a constant size fooling set first and then prove that one of size \( n \) exists for any given \( n \) which is basically the same as finding an infinite fooling set.

Note that another method to prove non-regularity is via reductions. Suppose you want to prove that \( L \) is non-regular. You can do regularity preserving operations on \( L \) to obtain a language \( L' \) which you already know is non-regular. Then \( L \) must not have been regular. For instance if \( \overline{L} \) is not regular then \( L \) is also not regular. You will see an example in Problem 4 below.

Prove that each of the following languages is not regular.

1. \( \{\theta^{2n}1^n \mid n \geq 0\} \)

2. \( \{\theta^m1^n \mid m \neq 2n\} \)

3. \( \{\theta^n \mid n \geq 0\} \)

4. Strings over \( \{\theta, 1\} \) where the number of \( \theta \)s is exactly twice the number of 1s.

   - Describe an infinite fooling set for the language.
   - Use closure properties. What is language if you intersect the given language with \( \theta^*1^* \)?

5. Strings of properly nested parentheses \( ( ) \), brackets \( [ ] \), and braces \( \{ \} \). For example, the string \( ([])\} \) is in this language, but the string \( ([]\}) \) is not, because the left and right delimiters don’t match.

   - Describe an infinite fooling set for the language.
   - Use closure properties.

6. \( w \), such that \( |w| = \lceil k\sqrt{k} \rceil \), for some natural number \( k \).

   Hint: since this one is more difficult, we’ll even give you a fooling set that works: try \( F = \{\theta^m \mid m \geq 1\} \). We’ll also provide a bound that can help: the difference between consecutive strings in the language, \( \lceil (k + 1)^{1.5} \rceil - \lceil k^{1.5} \rceil \), is bounded above and below as follows:

   \[
   1.5\sqrt{k} - 1 \leq \lceil (k + 1)^{1.5} \rceil - \lceil k^{1.5} \rceil \leq 1.5\sqrt{k} + 3
   \]

   All that’s left is you need to carefully prove that \( F \) is a fooling set for \( L \).

7. Strings of the form \( w_1#w_2#\cdots#w_n \) for some \( n \geq 2 \), where each substring \( w_i \) is a string in \( \{\theta, 1\}^* \), and some pair of substrings \( w_i \) and \( w_j \) are equal.
Work on these later:

7. \( \{ \mathbf{0}^n^2 \mid n \geq 0 \} \)

8. \( \{ w \in (0 + 1)^n \mid w \text{ is the binary representation of a perfect square} \} \)