1. Exercise 7.10 of the CC Book: Let \( \text{match} \) be the function that accepts a \( 3m \)-bit string \( x \) and an \( m \)-bit string \( y \) and returns 1 iff \( y \) is a substring of \( x \). Prove that \( D^\text{best}(\text{match}) = \Omega(m) \).

2. Exercise 7.11 of the CC Book: Let \( \text{sum}(a, b, i) \) be the function that takes two \( n \)-bit integers \( a, b \) and a \( \log n \)-bit integer \( i \) and returns the \( i \)-th bit of the binary representation of the sum \( a + b \) (the length of the input is \( m = 2n + \log n \)). Similarly, let \( \text{prod}(a, b, i) \) be the function that takes the same inputs and returns the \( i \)-th bit of the product \( a \cdot b \). Prove

   1. \( D^\text{best}(\text{sum}) = O(\log m) \); but
   2. \( D^\text{best}(\text{prod}) = \Omega(m / \log m) \).

3. Exercise 12.6 of the CC Book: In this exercise we are concerned with finite automata. Those are similar to Turing machines but they have only an input tape (and no read/write tapes) and the head is only allowed to move one cell to the right at each step. It is well known that any nondeterministic automaton with \( k \) states can be transformed into a deterministic one that has \( 2^k \) states. For some constant, consider the (finite) language

   \[ L_c = \{ ww \mid w \in \{0, 1\}^c \}. \]

   (1) Prove that there is a co-nondeterministic automaton with \( O(c) \) states that accepts the language \( L_c \). (2) Use communication complexity to prove that any deterministic automaton that accepts the language \( L_c \) requires at least \( 2^c \) states. Conclude that the above-mentioned transformation from deterministic automata to nondeterministic automata is optimal.