1. Let \( f, g : \mathbb{N} \to \mathbb{N} \) be two functions. Recall that \( f = O(g) \) if there exists a \( c > 0 \) such that \( f(n) \leq c \cdot g(n) \) for every sufficiently large \( n \). We say that \( f = \Omega(g) \) if \( g = O(f) \) and that \( f = \Theta(g) \) if \( f = O(g) \) and \( g = O(f) \). Also, we say that \( f = o(g) \) if for any \( \varepsilon > 0 \), \( f(n) \leq \varepsilon \cdot g(n) \) for every sufficiently large \( n \). Finally, we say that \( f = \omega(g) \) if \( g = o(f) \).

Prove or disprove:

(a) \( (5n)! = O(n^5) \).

(b) If \( f(n) = O(n) \) then \( 10^{f(n)} = O(2^n) \).

(c) \( \log(n!) = \Theta(n \log n) \).

(d) Every two functions \( f, g \) satisfy \( f = O(g) \) or \( g = O(f) \).

(e) There exists a function \( f \) such that \( f(n) = O(n^{1+\varepsilon}) \) for any \( \varepsilon > 0 \) but \( f(n) = \omega(n) \).

2. Show that any 1-tape Turing machine which at any step may move to the left, to the right or stay, can be simulated by a standard 1-tape Turing machine where the head has to move at any step. Compare the time complexity and the space complexity of the two machines.

3. Prove or disprove:

If a language \( A \) is decided by a 1-tape Turing machine \( M \) with at most \( 100 \cdot n^5 \) steps on an input of size \( n \) then there exists a 1-tape Turing machine \( M' \) that decides \( A \) with at most \( n^5 \) steps on an input of size \( n \).

4. Prove that each of the following problems can be solved by a polynomial time algorithm:

(a) Input: A graph \( G \) and a positive integer \( k \).

   Question: Does \( G \) contain a vertex of degree at least \( \log_2 |V(G)| \) or a clique of size \( k \)?

   \( |V(G)| \) denotes the vertex set of \( G \).

(b) Input: A 3CNF formula \( \phi \) in which each clause contains exactly 3 distinct literals and each variable occurs exactly 3 times.

   Question: Is \( \phi \) satisfiable?

   Hint: Use the fact that any regular bipartite graph has a perfect matching.\(^1\)

\(^1\)A regular graph is a graph where each vertex has the same number of neighbors. A matching in a graph is a set of edges without common vertices. A perfect matching is a matching which matches all vertices of the graph.