In the present work we study the first polynomial algorithm, which tests if the given word is a fixed point of a nontrivial morphism. This work contains an improved worst-case complexity estimate $O(m \cdot n)$ where $n$ denotes the word length and $m$ denotes the size of the alphabet. In the second part of this work we study the union-find problem, which is the crucial part of the described algorithm, and the Ackermann function, which is closely linked to the union-find complexity. We summarize several common methods and their time complexity proofs. We also present a solution for a special case of the union-find problem which appears in the studied algorithm. The rest of the work focuses on a Java implementation, whose time tests correspond to improved upper bound, and a visualization useful for particular entries.