In the years following the seminal paper *Go To statement considered harmful* (1968), by Dijkstra, structured programming became a major issue in computer science and was the center of discussions among eminent researchers as Dijkstra, Knuth, Hoare, Dahl, among others. With the time passing, structured programming became almost a standard technique for designing algorithms. Basically, an algorithm so designed is restricted to employ certain rules and statements, which makes the final formulation much easier to analyze and follow. Looking back at these developments, we observe that an important issue of structured programming has been left unsolved: Given an arbitrary algorithm $A$, is it structured? That is, whether or not $A$ has been written according to the rules of structured programming. The natural tool to tackle this problem is to analyze the flow graph of $A$. In fact, we propose a solution of it employing graph theory and classical algorithmic techniques, starting from $A$. It leads to a polynomial time algorithm to recognize whether or not a given algorithm is structured. Applications of it are also discussed.