Problem

Consider a directed acyclic graph (DAG) $G = (V_G, E_G)$, where V_G is a set of vertices and E_G is a set of directed edges. A sub-DAG $H = (V_H, E_G)$ of G is called consistent if it has the following property: $V_H \subseteq V_G$ and for any $v \in V_H$ all of the parents of v must also be in V_H.

Develop and implement an algorithm that enumerates all consistent sub-DAGs for a given DAG G. You can assume the DAG has a single root node (node without any parents). Your algorithm must be sub-exponential in the number of nodes V_G; that is, it is not allowed to generate all subsets of the set of nodes V_G and then check if each such subset is consistent. However, you can use this brute-force algorithm to test your new algorithm on various small problems.

a) Special case: The DAG G is a tree. That is, each node can have at most one parent, but it can have any number of children.

b) General case: The DAG G is not a tree. That is, each node can have any number of parents and any number of children, as provided by the data.

Develop a small example in which you can count the sub-DAGs manually. Visualize this DAG and use it to explain your algorithm. Demonstrate (or better, prove) correctness of your approach and then implement your algorithm and apply on the data sets provided to achieve counts. Your program should be able to accept an input of the type we provided in the data sets so that it can be tested independently. You can use any programming language you are comfortable with.