PAL labs 4

12 / 10 / 2022

You have to find and write out all paths of length 3 in a simple (no parallel edges) directed acyclic graph. What is the maximum possible number of these paths relatively to the number of nodes in the graph? What is the asymptotic complexity of your algorithm?

Find an oriented graph such that all input and output degrees of each node is at least 1 and there is at least one node in the graph which does belong to any cycle.

We construct a graph of type (r) in the following way: let's have two disjunctive sets of nodes $A = \{a_1, a_2, ...a_r\}$, $B = \{b_1, b_2, ...b_r\}$; further let's create a complete graph over the set A, and a complete graph over the set B. Finally, we add edge between edges of the nodes with the same indices, i.e. $(a_1, b_1), (a_2, b_2)..., (a_r, b_r)$. For which r will the final graph be an Euler one?

A weakly directed graph G with n nodes and m edges contains c_1 roots and c_2 leaves, while n, c_1, and c_2 are given. For which values of m it is ensured that the graph G will be acyclic? What is the maximal possible value of m given n, c_1, c_2?

We say that a directed graph is homogenic if the distance (= the shortest edge-wise path) of each tuple (root, leaf) is the same across all tuples. Design an effective algorithm which decides if a given graph is homogenic or not, and compute its asymptotic complexity. Is it possible to make the algorithm faster if we know that the graph is acyclic?

What is the highest possible degree of an node (degree = number of sons) in a binomial heap with N keys?

A node in a binomial heap can have a degree (= number of sons) higher than 2. The node points at other binomial subtrees. We have two options:

a) pointers are ordered in the increasing order of subtrees sizes, to which the node points

b) the pointers are ordered in randomly

Decide if the choice of a) or b) affects the speed of *Insert* and *DeleteMin* operations.

In a binomial heap, which keeps minimal values in the roots of its trees, we are asked to find a key with the maximal value and remove it from the heap. What is the asymptotic complexity of this process?

Let's assume a binomial heap H with k binomial trees $T_1, T_2, \ldots T_k$. How many leaves are in the heap H? If H contains n keys, what is the maximal possible value of k w.r.t. n?