PAL labs 3

5 / 10 / 2022

Let's assume we have an undirected simple graph $G = (V, E)$, which is represented by a list of edges (which is not sorted in any way) and the access to and edges is sequential. What is the asymptotic complexity of BFS and DFS?

Assuming a graph with n nodes and $O(n^2)$ edges, and a constant access time to each edge and node, then the asymptotic complexity of DFS is $O(n^2)$. Show the asymptotic complexity of DFS in the case when access time of each node and each edge is in $O(n^{\frac{1}{2}})$.

A set of edges, i.e. tuples $(node, node)$ is stored in a list. We know, the graph has N nodes and that it is disconnected. It also contains a component K with at least $N/2 + 1$ nodes. We are asked to create a new list containing only edges corresponding to the component K. Describe and algorithm solving this task, possible as fast as possible. What is its asymptotic complexity? The order of edges in both lists does not care.

We need to find a spanning tree of the graph, not necessarily minimal, with the requirement that the price of each edge of the searched spanning tree must be in the interval $< c_1, c_2 >$.
Is it necessary to use an algorithm for finding minimum spanning trees, or just a simple procedure?

Let's have an undirected weighted simple graph G represented by a weight matrix C. What is the asymptotic complexity of Kruskal's algorithm for searching minimum spanning tree assuming the access time of a single element in C is a constant, and each single operation of *Find* and *Union* is proportional to the number of nodes in G?

On your computer, you have to find the minimum spanning tree of a weighted complete graph. Estimate what is the order of maximum number of nodes such a graph can have to solve the task through the night until the next day.

We found the minimum spanning tree of a graph supplied with many millions of vertices and transferred it to the customer. In the afternoon the customer calls that the graph specification contains an error and that in fact the edge between vertices 2075154 and 11439446 is about 17% cheaper than as described in the original specification. All graph data and the minimum spanning tree are still on our disk. We should determine in linear time with regard to the number of vertices if this change affects the shape and the price of the minimum spanning tree, and if so, to give the shortest possible fix that can convey back using the phone.

We say that two directed graphs are weakly equivalent iff their respective condensations contain equal number of nodes. What is the best possible asymptotic complexity of verification whether two graphs are weakly equivalent?

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?