Ex # 3.

Topic: Graph algorithms

Requirements:

1. Definition of directed and undirected graph.

2. Knowledge of the following graph representations in a digital machine:

- neighborhood matrix (vertex matrix)

- incident matrix (egde matrix)

- edge list

- list of incidents

and knowledge of the memory complexity for these representations and the time complexity of obtaining information about the vertices and edges of the graph for each of these representations.

3. Knowledge of the procedure of visiting the graph in depth and breadth (DFS and BFS)

4. Knowledge of the topological sorting algorithm and its computational complexity.

Instruction

For a randomly generated undirected input graph (the input graph is loaded from a text file), masure the time of obtaining information about the existence of edge between a pair of random vertices, for each of graph representations listed above. Perform tests for all elements randomly selecting individual vertices. Calculate the average search time (by adding up the search times for individual elements and dividing this sum by the number of elements) (number of vertices in the graph - n, saturation of the graph with edges 0.6). Graph t = f (n) for each representation; on one chart, for each search.

2. For a randomly generated DAG graph (n - vertices and edge fill factor 0.3), use the topological sorting procedure. Chart t = f (n) - time to create the sorted sequence. Justify the choice of graph representation. Give the advantages and disadvantages of the selected representation in comparison with the others.

3. Formulate conclusions regarding the effectiveness of each graph representation and the problem of topological sorting.

Duration: 2 weeks