

Algorithms

Lab 8 (8% + 4% bonus)

Topics: Graph Algorithms;

Textbook: Cormen Introduction to Algorithms 3rd ed.

Problem 01 (2%)

Depth-First Search Algorithm (DFS)

1. Read Cormen 21.1, 22.3, Appendix B.4
2. Use this algorithm to implement following C++ program:
Given size of undirected unweighted graph and adjacency matrix describing it. Program has to output the number of connected components in this graph and their descriptions. Description of a connected component is a set of component's vertexes in arbitrary order. Concrete format of input and output is your choice. Use adjacency lists to represent graph in your program.

Problem 02 (2%)

Topological Sorting

1. Read Cormen 21.1, 22.3, 22.4, Appendix B.4
2. Use this algorithm to implement following C++ program:
Given N - the number of all courses student has to take and M – number of pairs describing prerequisites. For example, pair “2 3” means that student can take course 3 only after course 2. Your program has to output sequence of all courses representing the order which student follow in his educational track without violation of prerequisites. Concrete format of input and output is your choice.

Problem 03 (2%)

Breadth-First Search Algorithm (BFS)

1. Read Cormen 21.1, 22.2, Appendix B.4
2. Use this algorithm to implement following C++ program:
Given size of undirected weighted graph, adjacency matrix describing it and source vertex. Program has to output shortest paths to all vertexes of the graph from source vertex. Concrete format of input and output is your choice. Use adjacency lists to represent graph in your program.

Problem 04 (2%)

Bellman-Ford Algorithm

1. Read Cormen 24.Intro, 24.1, Appendix B.4
2. Use this algorithm to implement following C++ program:
Given list of edges of directed weighted graph and source vertex. If graph has a cycle of negative weight output corresponding message, otherwise output shortest paths to all vertexes from source vertex. Concrete format of input and output is your choice.

Bonus: Problem 05 (4%)

Minimum Spanning Tree. Prim's Algorithm.

1. Read Cormen 23.1, 22.2, 6.1 < Appendix B.4
2. Use this algorithm to implement following C++ program:
Given list of edges of undirected weighted connected graph. Program has to output the weight of its minimum spanning tree. Concrete format of input and output is your choice. Implement min-priority queue as a binary min-heap (6.1).