FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 5

“Algorithms on graphs. Introduction to graphs and basic algorithms on graphs”

Performed by

*Pakulev Aleksandr*

*J4133c*

Accepted by

Dr Petr Chunaev

St. Petersburg

2021

**Goal:**

The use of different representations of graphs and basic algorithms on graphs (Depth-first search and Breadth-first search)

**Tasks:**

1. Generate a random adjacency matrix for a simple undirected unweighted graph with 100 vertices and 200 edges (note that the matrix should be symmetric and contain only Os and Is as elements). Transfer the matrix into an adjacency list. Visualize the graph and print several rows of the adjacency matrix and the adjacency list. Which purposes is each representation more convenient for?
2. Use Depth-first search to find connected components of the graph and Breadth-first search to find a shortest path between two random vertices. Analyse the results obtained.
3. Describe the data structures and design techniques used within the algorithms.

**Brief information:**

1. Dfs – the simplest algorithm on graph, which may solve a wide range of different tasks, such as find connected components, topology graph sort and e.t.c. time complexity O(N + M).
2. Bfs algorithms required to determine minimum way between two random nodes
3. BFS – the simplest algorithm on graph too, this algorithm may find the shortest way in graph, each edge's weight is one between two nodes. time complexity O(N + M).

**Result:**

Adjacency matrix and list were generated.

__ 00 01 02 03 04 05 06 07 08 09

00 00 01 00 01 00 01 00 01 00 01

01 01 00 00 00 00 00 00 00 00 00

02 00 00 00 00 00 00 00 00 00 00

03 01 00 00 00 00 00 00 00 00 00

04 00 00 00 00 00 00 00 00 00 00

05 01 00 00 00 00 00 00 00 00 00

06 00 00 00 00 00 00 00 00 00 00

07 01 00 00 00 00 00 00 00 00 00

08 00 00 00 00 00 00 00 00 00 00

09 01 00 00 00 00 00 00 00 00 00

adjacency matrix

00->01 03 05 07 09

01->00

02->

03->00

04->

05->00

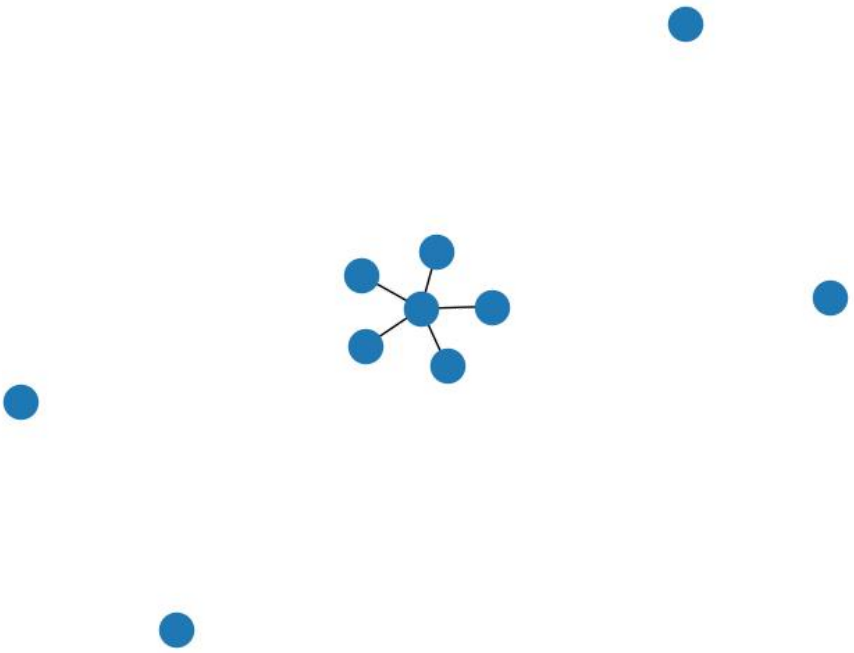
06->

07->00

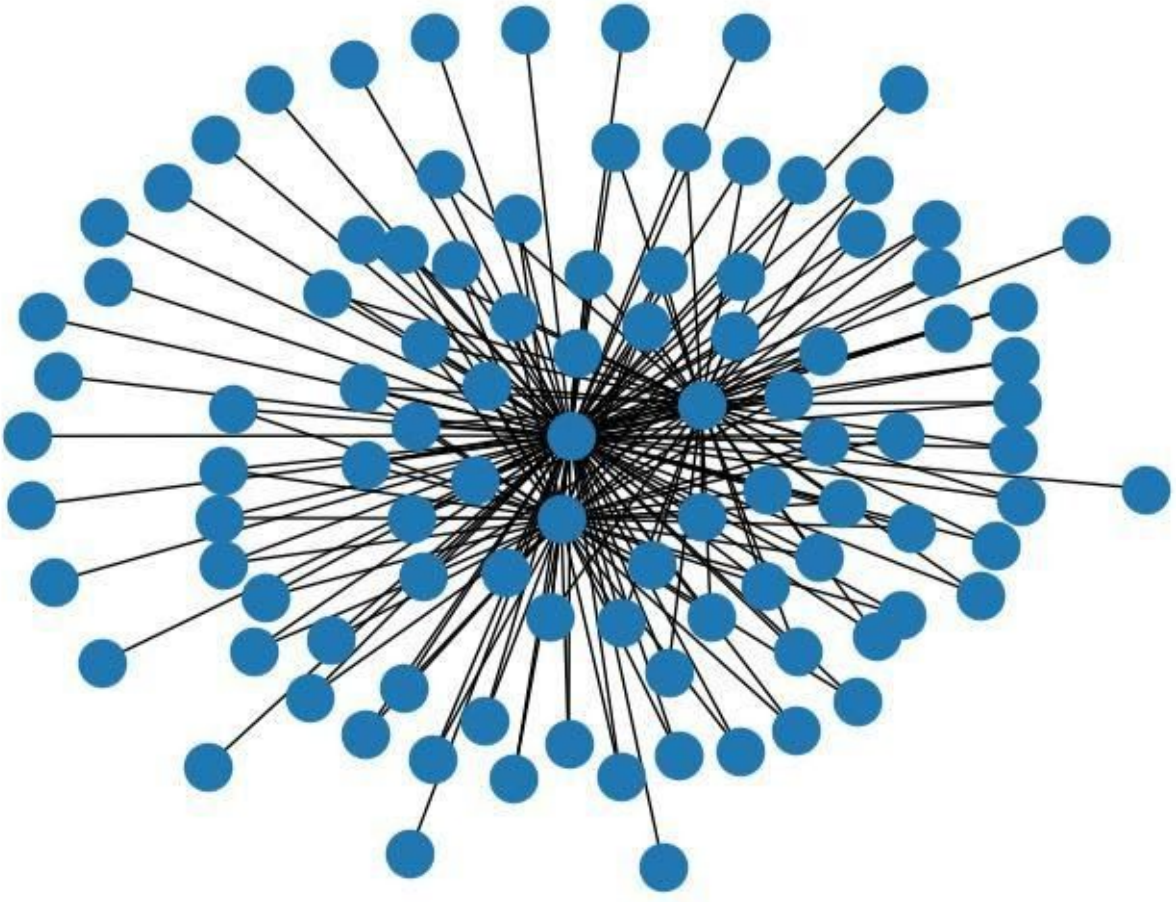
08->

09->00

Corresponded adjacency list



Graph with 10 nodes and 5 edges



Graph with 100 nodes and 200 edges

Adjacency list:

1) As usual less memory consumption. So better to use them
2) Better to use for sparse graphs.

3) May support representation when between the same nodes exists more then one edges.

Adjacency matrix:

1) Better to use when graph is comprehensive.
2) Typically, useless.

Use dfs find connected component. For graph <100, 200> above Only single connected component, it's means graph is connected.

Use bfs find the shortest way between two vertex, analyses result.

At graph above <100, 200> distance between vertex 1 and 5 is 2(two).

This (BFS) algorithm is greedy algorithm, because, edges weight is constant so when we will found distance for vertex TO we can't update vertex which we still visited.

DFS is greedy too, just because we assume that the best answer may be received just in one step deeper.

**Conclusion:**

In this task I implemented dfs and bfs algorithms which need to 1) answered which property graph has, such as it's connected or not. 2) find the shortest way between to vertexes, or say that nodes are in different components.