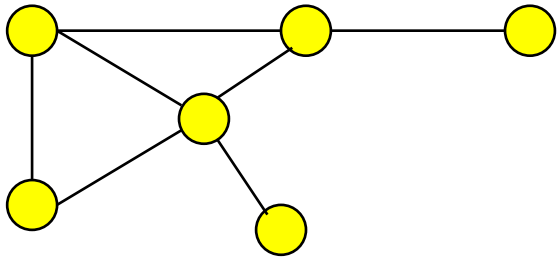
Exercise #6	Algorithms and Data Structures	
	Topic: Dynamic data structures – graphs	Version: 1.0 / 2019
	Prepared by: dr inż. Grzegorz Łukawski & dr inż. Barbara Łukawska	

1) Graphs

1.1) The definition

Graph – a data structure consisting of finite number of nodes (vertices), connected with edges (pointers). Graph doesn't limit the uniqueness of paths. Graph is a generalization of other dynamic data structures, such as lists and trees. Example representation of a graph:



1.2) Types of graphs

- Undirected graph consists of finite set of nodes N and finite set of edges E. Each edge is an unordered pair of nodes.
- Directed graph is a graph with directed edges, usually drawn as arrows. Each edge is an ordered pair of nodes, where edge from A to B is not equal to edge from B to A.
- Graph with weights is a graph, where a weight is associated with each edge. May be undirected or directed.

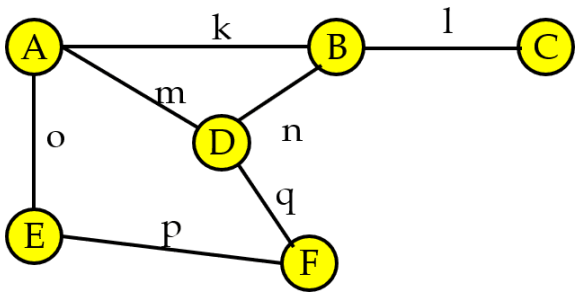
1.3) Representation of graphs in memory

Three methods of representing a graph in memory are used in practice. These are the following:

- Adjacency list – every node stores a list of other connected nodes;
- Adjacency matrix – a matrix with rows and columns representing nodes;
- Incidence matrix – a matrix in which rows represent nodes and columns represent edges.

1.4) Example graph

Undirected graph with 6 nodes and 7 edges:



Represented as the adjacency list:

A → B → D → E
B → A → C → D
C → B
D → A → B → F
E → A → F
F → D → E

Represented as the adjacency matrix (empty cells store zero):

	A	**B**	**C**	**D**	**E**	**F**
A		1		1	1	
B	1		1	1		
C		1				
D	1	1				1
E	1					1
F				1	1	

Represented as the incidence matrix (empty cells store zero):

	k	**l**	**m**	**n**	**o**	**p**	**q**
A	1		1		1		
B	1	1		1			
C		1					
D			1	1			1
E					1	1	
F						1	1

---

2) Exercises

Write a program in **C++**, storing a fixed-size undirected graph (at least 5 nodes) as an adjacency matrix (in form of a static two-dimensional array). The program should have the following features:

A) Adding a new edge to the graph (between given nodes), the actual number of nodes should be determined on the basis of maximum node number entered by the user (up to the maximum mentioned above);

B) Displaying the graph as an adjacency matrix and adjacency list;

Additional exercises:

C) Repeat exercises A and B for directed graph.