Notations

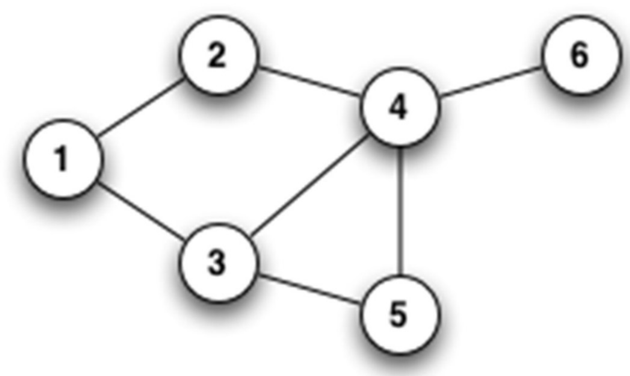
Graph Representation

There are two classical ways to represent a graph: an adjacency matrix, or a set of adjacency lists. The choice of the type of representation depends on the operations performed on this structure:

- for the representation by the adjacency matrix, the verification of the existence of an arc between two given vertices is in O(1), whereas the search for the neighbours of a given vertex is in O(n).

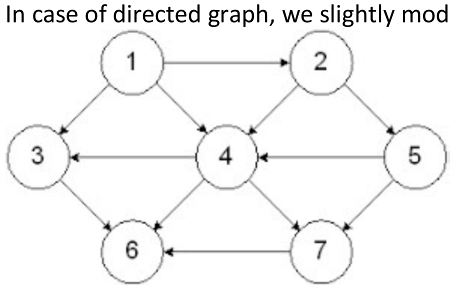
- for the representation by the adjacency list, the verification of the existence of an arc between two given vertices is in O(n), whereas the search for the neighbours of a given vertex is in O(1).

In our case, we have opted for the adjacency list representation because the most common operation is the traversal of the list of neighbours.



The above undirected graph will be represented by the following list:
L= {(2, 3), (1, 3, 4), (1, 4,5), (2, 3, 5, 6), (3, 4), (4)}
L[1]=(2, 3) is the nodes connected to node 1.

In case of directed graph, we slightly modified the adjacency list representation.



The above directed graph will be represented by the following list:
L = {((), (2, 3, 4)), ((1), (4, 5)), ((1, 4), (5)), ((1,2,5), (3, 6, 7)), ((2), (4, 7)), ((4, 3, 7), ()), ((4, 5), (6))}
L[2][0]=(1) are the incoming nodes to node 2.
L[2][1]=(4, 5) are the outgoing nodes to node 2.

Notations

- Let G= (V, E) be a graph with n = |V| vertices and m = |E| edges.
- S = Set of source nodes
- T = Set of terminal nodes
- IN_N(n) represents the set of incoming nodes to node n
- OUT_N(n) represents the set of outgoing nodes to node n
- L[n] represents the set of nodes connected to n