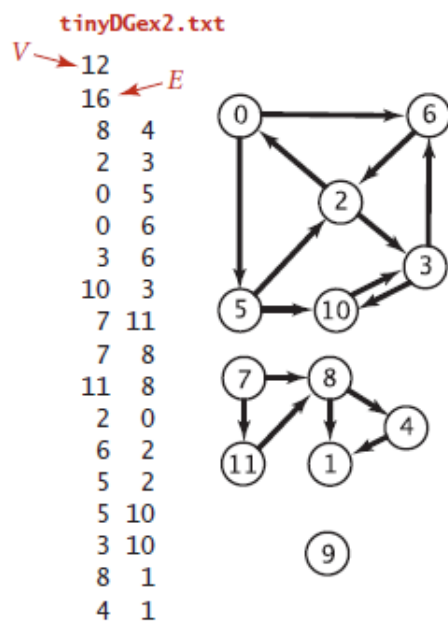


Grafet e Drejtuar

- Shpjegoni termat e mëposhtëm për një graf të drejtuar:
 - outdegree
 - indegree
 - strong component
 - DAG
 - Topological sort
 - KosarajuSharir algorithm
- What is the maximum number of edges in a digraph with V vertices and no parallel edges?
What is the minimum number of edges in a digraph with V vertices, none of which are isolated?
- Draw the adjacency lists built by using Digraph's input stream constructor for the file presented in the figure below.



- The *indegree* of a vertex in a digraph is the number of directed edges that point to that vertex.
 - The *outdegree* of a vertex in a digraph is the number of directed edges that emanate from that vertex.
 - No vertex is reachable from a vertex of outdegree 0, which is called a *sink*;
 - A vertex of indegree 0, which is called a *source*, is not reachable from any other vertex.
 - A digraph where self-loops are allowed *and* every vertex has outdegree 1 is called a *map* (a function from the set of integers from 0 to $V-1$ onto itself).

Write a program Degrees.java that implements the following API:

<code>public class Degrees</code>		
	<code>Degrees(Digraph G)</code>	<i>constructor</i>
	<code>int indegree(int v)</code>	<i>indegree of v</i>
	<code>int outdegree(int v)</code>	<i>outdegree of v</i>
	<code>Iterable<Integer> sources()</code>	<i>sources</i>
	<code>Iterable<Integer> sinks()</code>	<i>sinks</i>
	<code>boolean isMap()</code>	<i>is G a map?</i>

5. What are the strong components of a DAG?
6. What happens if you run the Kosaraju–Sharir algorithm on a DAG?
7. The reverse postorder of a digraph’s reverse is the same as the postorder of the digraph.
8. True or false: If we modify the Kosaraju–Sharir algorithm to run the first depthfirst search in the digraph G (instead of the reverse digraph G^R) and the second depthfirst search in G^R (instead of G), then it will still find the strong components.
9. True or false: If we modify the Kosaraju–Sharir algorithm to replace the second depth-first search with breadth-first search, then it will still find the strong components.
10. *Directed Eulerian cycle.* A directed Eulerian cycle is a directed cycle that contains each edge exactly once. Write a Digraph client DirectedEulerianCycle that finds a directed Eulerian cycle or reports that no such cycle exists.

Hint : Prove that a digraph G has a directed Eulerian cycle if and only if G is strongly connected and each vertex has its indegree equal to its outdegree.
11. *Hamiltonian path in DAGs.* Given a DAG, design a linear-time algorithm to determine whether there is a directed path that visits each vertex exactly once.