Graphs and classic graph problems

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Encoding of graphs: edge lists, adjacency lists, node-edge incidence matrix adjacency matrix**

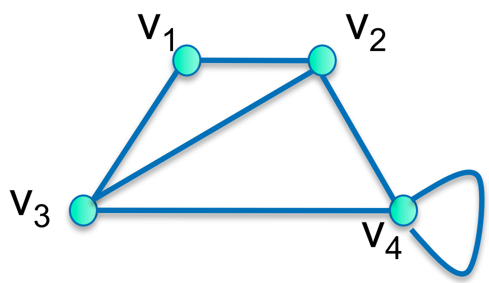
  – **edge list**
  - If G=(V,E) is a graph with n=|V| nodes and m=|E| edges, the **edge list** will look as follows:

    $n,m,\{a_1,e_1\},\{a_2,e_2\},\ldots,\{a_m,e_m\}$, with $\{a_i,e_i\}$ are end nodes of edge i

  - The sequential arrangement of edges within the list is arbitrary
  - {v,v} is inserted for a loop
  - In directed graphs, the first edge component marks the start node, the other one the end node

Graphs and classic graph problems

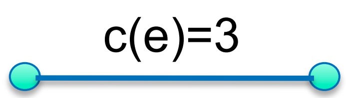
Example:



edge list: 4,6,$\{v_1,v_2\},\{v_2,v_4\},\{v_4,v_4\},\{v_3,v_4\},\{v_2,v_3\},\{v_3,v_1\}$

Graphs and classic graph problems

Edges may be supplied with weights or costs. I.e., there is a cost function c: $E \rightarrow \mathbb{Q}$



Encoding: weights for the edges are noted.

An edge list consumes 2m+2 memory cells for storeing,
an edge list with weights 3m+2 cells.

Graphs and classic graph problems

TECHNISCHE UNIVERSITÄT DARMSTADT

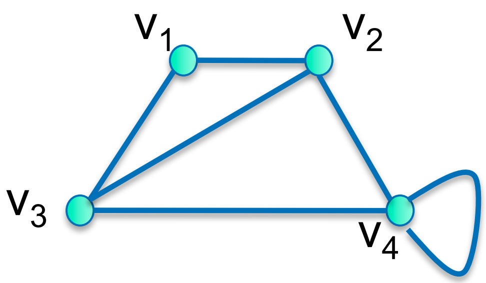
- **Graph coding: edge lists, adjacency matrizes, node-edge incidence-matrizes, adjacency lists**

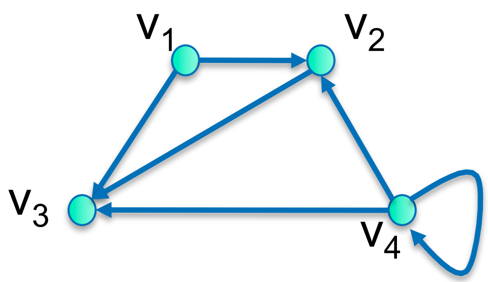
  – **Adjacency matrix**
  - Let G=(V,E) be a simple graph (i.e. no more than one edge per node pair) with nodes 1,...,n. Then, the matrix $A \in IR^{n \times n}$ with
    - A_{ij} = 1, if (i,j) is an edge,
    - A_{ij} = 0 , otherwise

    is called the **adjacency matrix** of G.
  - If A has edge weights, then we set $A_{ij} = c((i,j))$, if $(i,j) \in E$; and $A_{ij} = 0, +\infty$, oder $-\infty$ otherwise, depending on the task.
  - Memory consumptions is $O(n^2)$ cells.

Graphs and classic graph problems

Examples:





Adjacency matrix:

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Graphs and classic graph problems

TECHNISCHE UNIVERSITÄT DARMSTADT

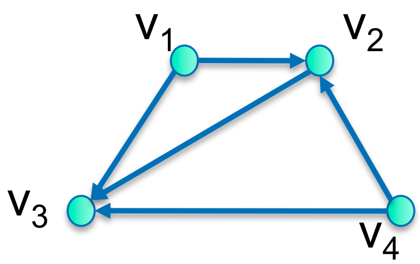
- Graph coding: edge lists, adjacency matrizes, node-edge incidence-matrizes, adjacency lists

  – **node-edge-incidence matrix**
  - Let G=(V,E) be a simple directed graph with the nodes 1,...,n. The matrix $A \in \{-1,0,1\}^{|V|\times|E|}$ with
    - A_{ij} = 1, if edge j leaves node i,
    - A_{ij} = -1, if edge j directs into node i,
    - A_{ij} = 0 otherwise

    is called **node-edge-incidence matrix** of G.
  - Memory consumption: $O(|V| \cdot |E|)$ cells.

Graphs and classic graph problems

Examples:



node-edge-incidence matrix:

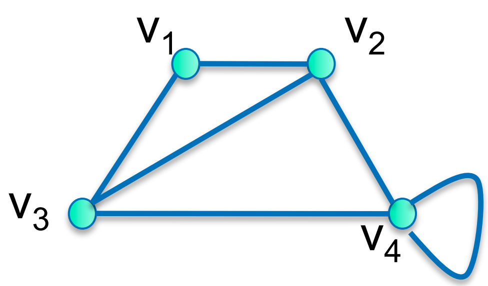
	$v_1 v_2$	$v_1 v_3$	$v_2 v_3$	$v_4 v_2$	$v_4 v_3$
v_1	1	1	0	0	0
v_2	-1	0	1	-1	0
v_3	0	-1	-1	0	-1
v_4	0	0	0	1	1

Graphs and classic graph problems

- **Graph coding: edge lists, adjacency matrizes, node-edge incidence-matrizes, adjacency lists**

  – **Adjacencyliste**
  When we store the number of nodes for a graph G=(V,E) plus the degree and the neighbours of each node , such a data structure will be called an adjecency-list of G.

  Memory consumtion: O(|V| + |E|).



#nodes	No. of node	degree	neighbour
4	1	2	2,3
	2	3	1,3,4
	3	3	1,2,4
	4	3	2,3,4