COMP2521 25T3

Graphs (IV)
Directed and Weighted Graphs

Sim Mautner

cs2521@cse.unsw.edu.au

directed graphs
weighted graphs

Generalising Graphs

In graphs representing real-world scenarios,
edges are often directional and may have a sense of cost.

Thus, we need to consider directed and weighted graphs.

Directed Graphs

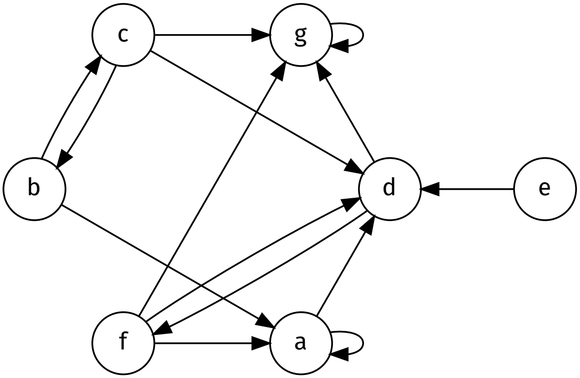
Some applications require us to consider directional edges: $v \to w \neq w \to v$
e.g., ‘follow’ on Twitter, one-way streets, etc.

In a directed graph or digraph:
edges have direction.

Each edge (v, w) has a source v and a destination w.

Directed Graphs

Example



application	vertex is...	edge is...
WWW	web page	hyperlink
chess	board state	legal move
scheduling	task	precedence
program	function	function call
journals	article	citation
make	target	dependency

Digraph Terminology (I)

in-degree

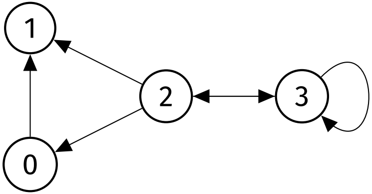
$\deg^{-}(v)$ or $\text{in}(v)$

the number of incoming edges to a vertex

out-degree

$\deg^{+}(v)$ or $\text{out}(v)$

the number of outgoing edges from a vertex



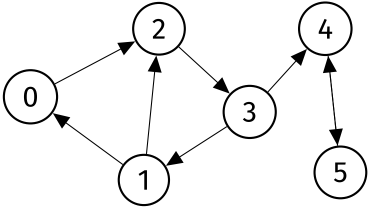
$$\begin{aligned} \text{in}(0) &= 1 & \text{out}(0) &= 1 \\ \text{in}(1) &= 2 & \text{out}(1) &= 0 \\ \text{in}(2) &= 1 & \text{out}(2) &= 3 \\ \text{in}(3) &= 2 & \text{out}(3) &= 2 \end{aligned}$$

Digraph Terminology (II)

A directed path is
a sequence of vertices where
each vertex has an outgoing edge to
the next vertex in the sequence

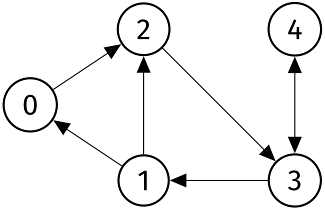
If there is a directed path from v to w,
then we say that w is reachable from v

A directed cycle is
a directed path where
the first and last vertices are the same
e.g., 0-2-3-1-0, 1-2-3-1

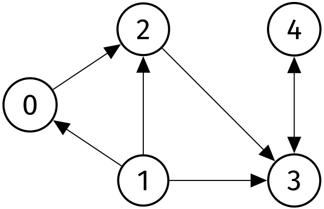


Digraph Terminology (III)

A digraph is strongly connected if
there is a directed path from every vertex to every other vertex



strongly connected

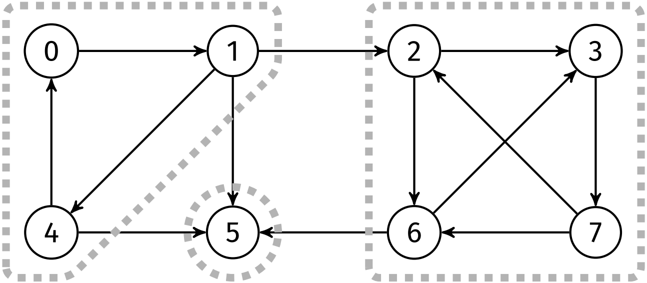


not strongly connected

Digraph Terminology (IV)

A strongly-connected component is a maximally strongly-connected subgraph.

A digraph that is not strongly connected has two or more strongly-connected components.



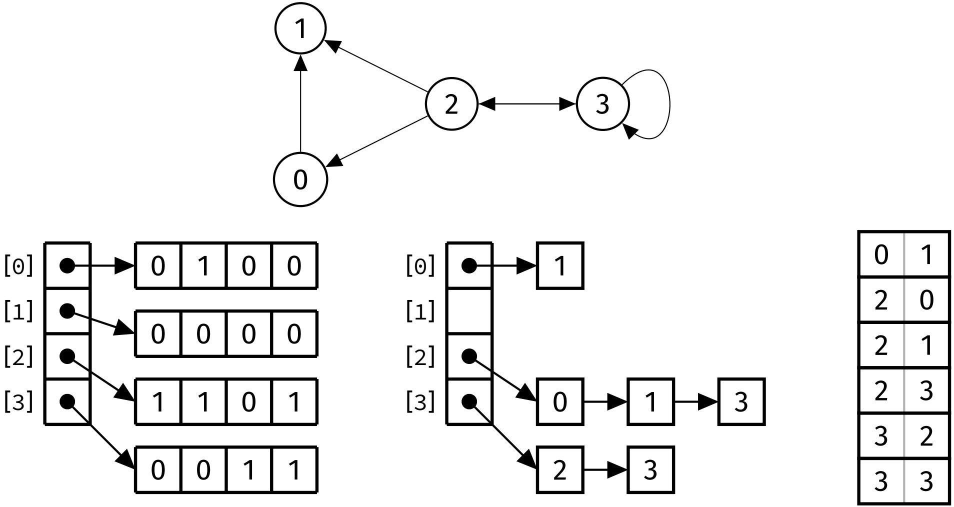
Directed Graphs
Representations

Same representations as for undirected graphs:

- Adjacency matrix
- Adjacency list
- Array of edges

Directed Graphs

Representations



Adjacency matrix

Adjacency list

Array of edges

Digraph Complexity

	Adjacency Matrix	Adjacency List	Array of Edges
Space usage	$O(V^2)$	$O(V+E)$	$O(E)$
Insert edge	$O(1)$	$O(\deg(v))$	$O(E)$
Remove edge	$O(1)$	$O(\deg(v))$	$O(E)$
Contains edge	$O(1)$	$O(\deg(v))$	$O(\log(E))$

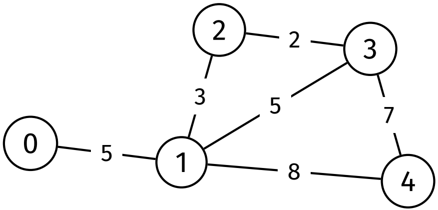
Real digraphs tend to be sparse (large V, small average $\deg(v)$),
so we use $\deg(v)$ to denote the degree of the source vertex v.

Weighted Graphs

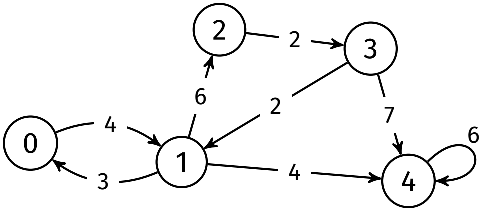
Weighted Graphs

Some applications require us to consider a cost or weight assigned to a relation between two nodes.

In a weighted graph, each edge (s, t, w) has a weight w.



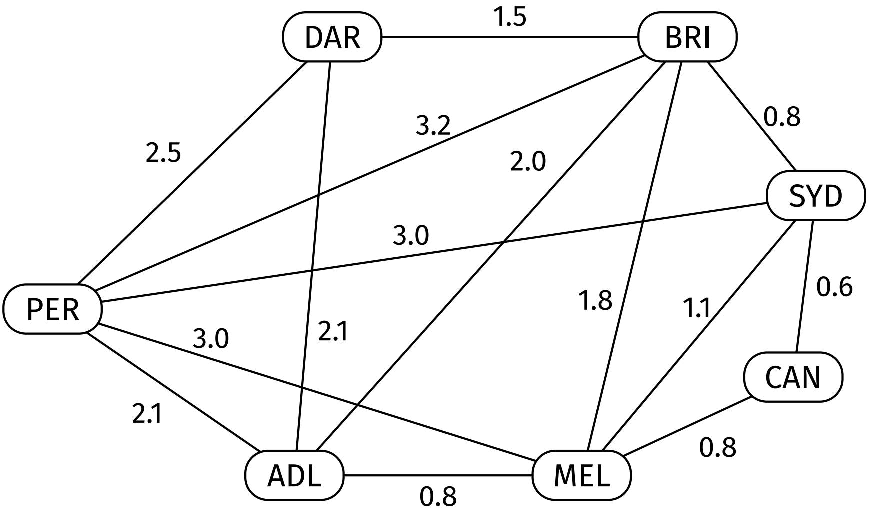
Weighted Graph



Directed Weighted Graph

Weighted Graphs

Example: Major airline routes in Australia



Weighted Graphs

Representations

Adjacency matrix:

- store *weight* in each cell, not just true/false
- need a value to signify "no edge"

Adjacency list:

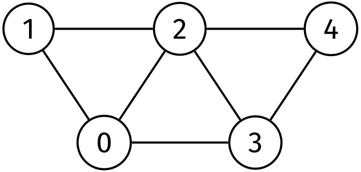
- add weight to each list node

Array of edges:

- add weight to each edge

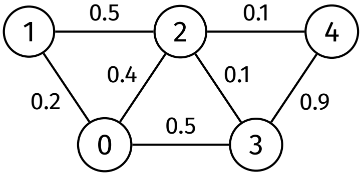
Weighted Graphs

Representations: Adjacency Matrix



$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

undirected, unweighted

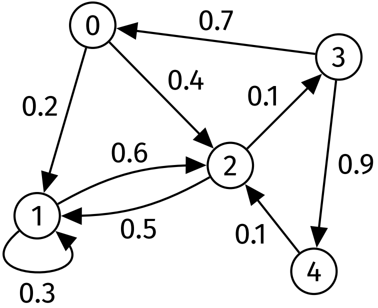


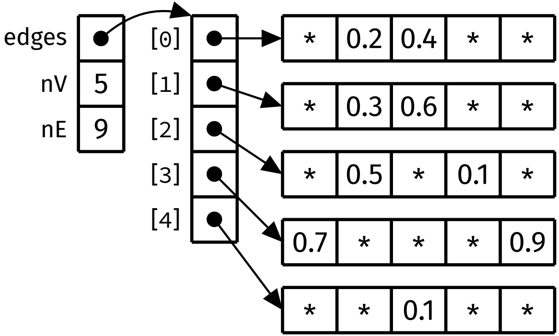
$$\begin{bmatrix} - & 0.2 & 0.4 & 0.5 & - \\ 0.2 & - & 0.5 & - & - \\ 0.4 & 0.5 & - & 0.1 & 0.1 \\ 0.5 & - & 0.1 & - & 0.9 \\ - & - & 0.1 & 0.9 & - \end{bmatrix}$$

undirected, weighted

Weighted Graph

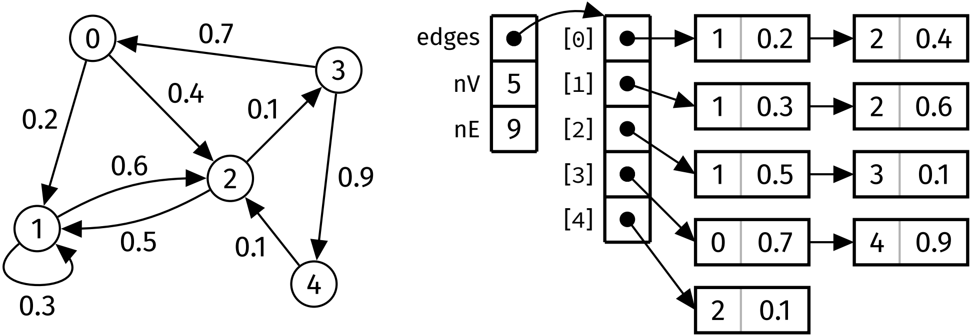
Representations: Adjacency Matrix





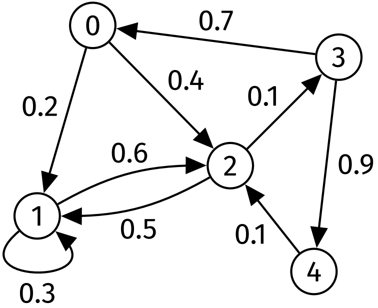
Weighted Graph

Representations: Adjacency List



Weighted Graph

Representations: Array of Edges



edges	•
nV	5
nE	9
maxE	···

0	1	0.2
0	2	0.4
1	1	0.3
1	2	0.6
2	1	0.5
2	3	0.1
3	0	0.7
3	4	0.9
4	2	0.1