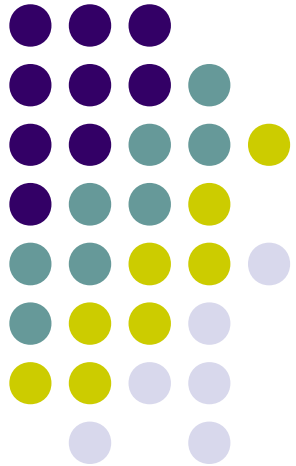
COMP20003
Algorithms and Data Structures
Introduction to Graphs

Nir Lipovetzky

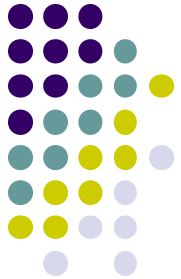
Department of Computing and Information Systems

University of Melbourne

Semester 2

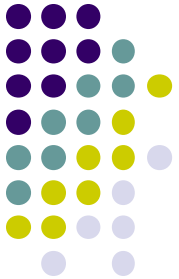


Graph definition



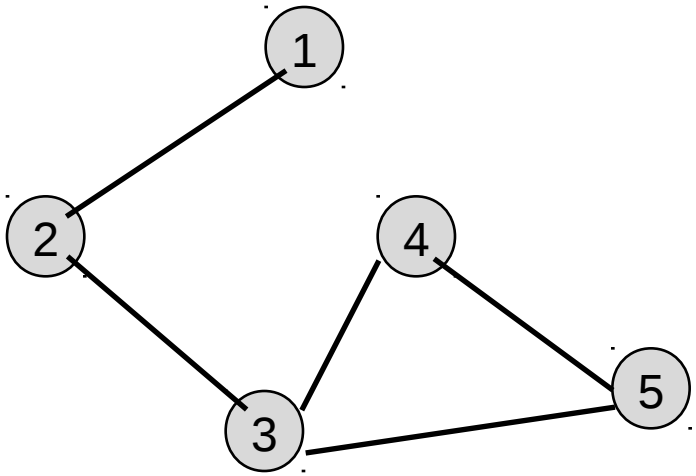
- Graph:
  - a representation of a set of objects
  - some pairs of objects are connected by links.

Graph definition



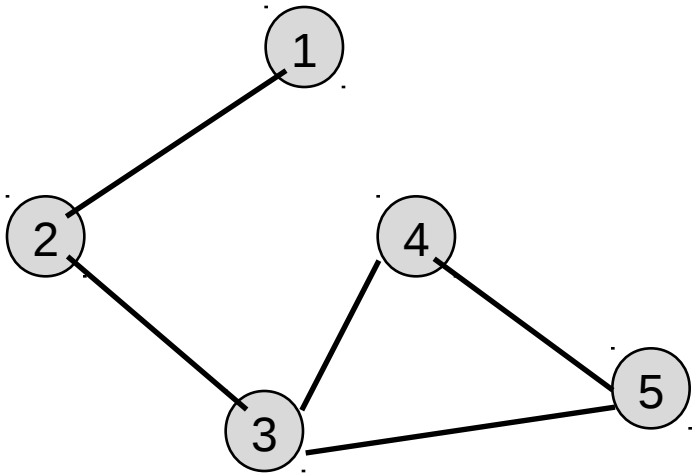
- Graph G = {V,E}
- Graph: a set of
  - Vertices V
  - Edges E (links between vertices)
- Comparing to trees, linked lists:
  - vertices = nodes
  - edges = links
- Vertices: can contain information
- Edges: can have direction and/or weight

Undirected graph



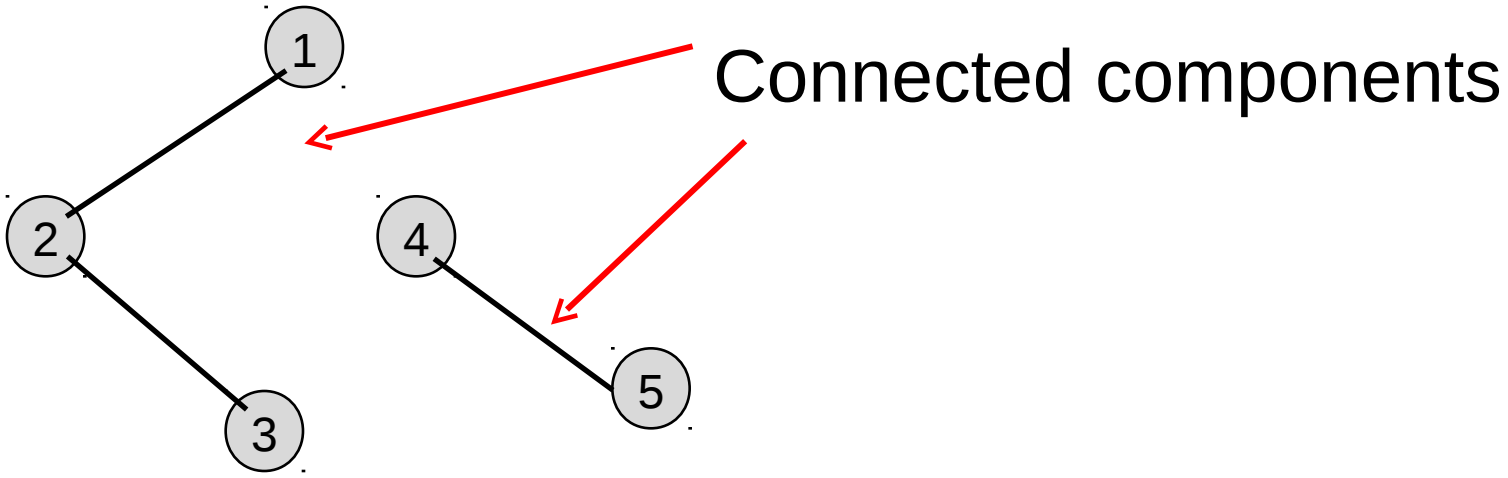
Edges have no direction specified.

Connected
Undirected graph

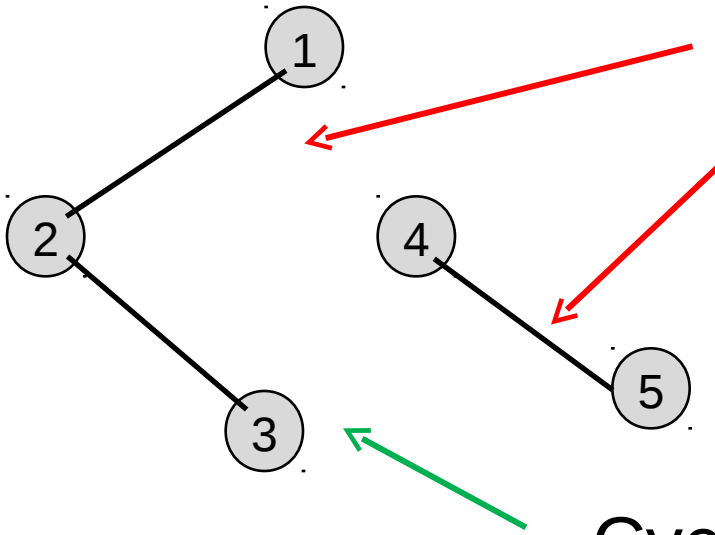


Every pair of vertices is connected (possibly indirectly).

Unconnected Undirected graph



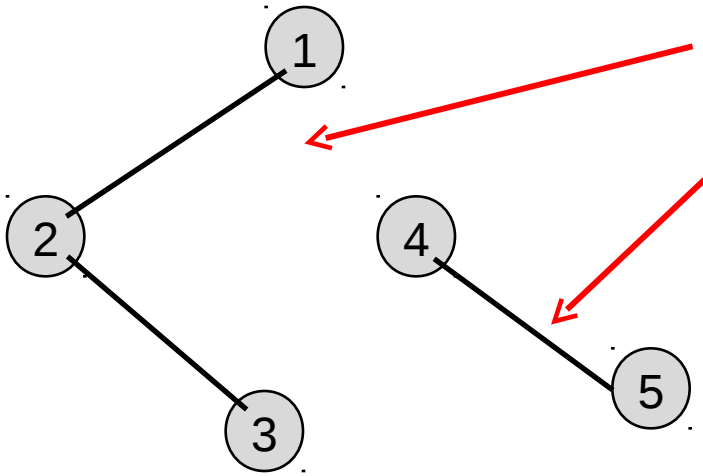
Unconnected Undirected graph with cycle



Connected components

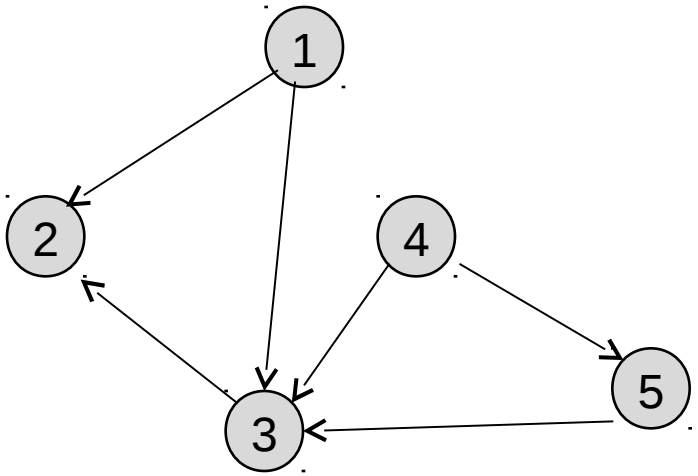
Cycle:
1→2→3→1, *etc.*

Acyclic, unconnected Undirected graph



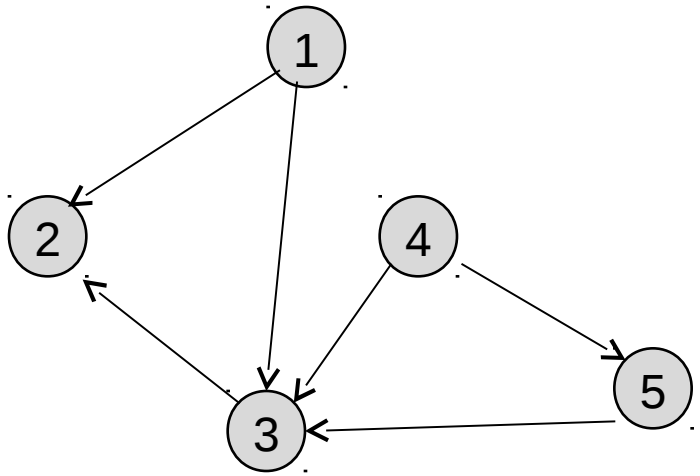
Connected components

Directed graph



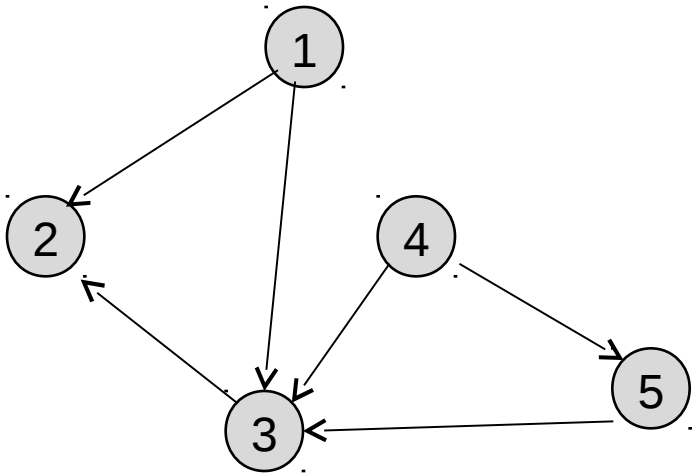
Edge direction is specified.
Links are not symmetrical.

Directed graph



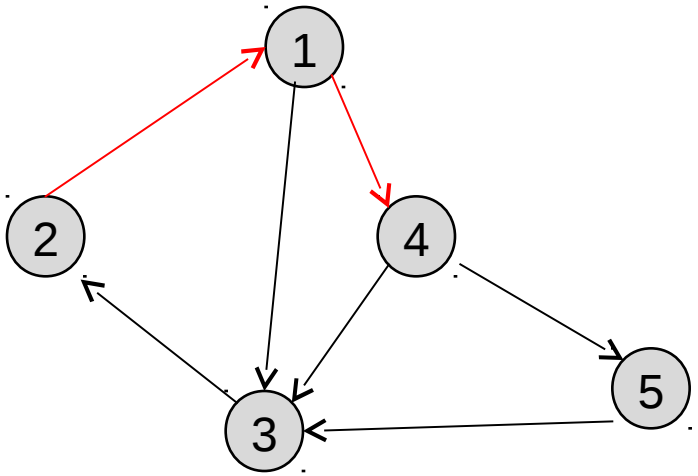
Reachability:
Can you get from Vertex 1 to Vertex 5?

Weakly connected Directed graph



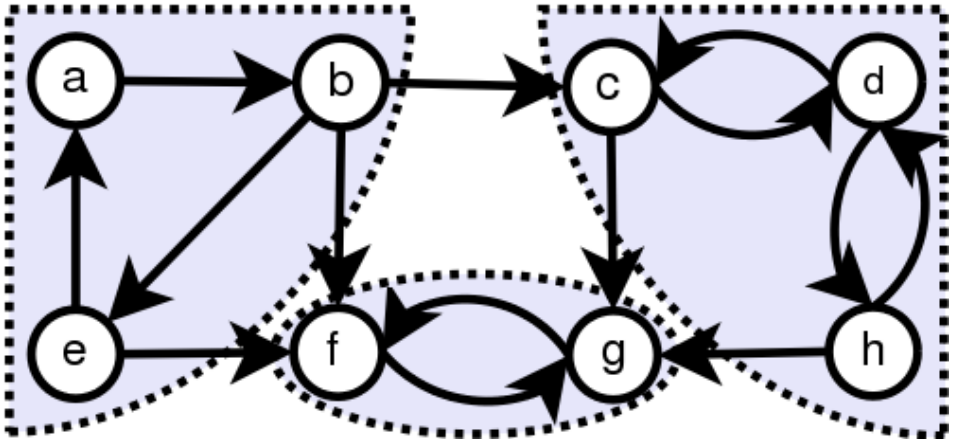
Replace all directed edges with undirected edges, to obtain a connected (undirected) graph

Strongly connected Directed graph



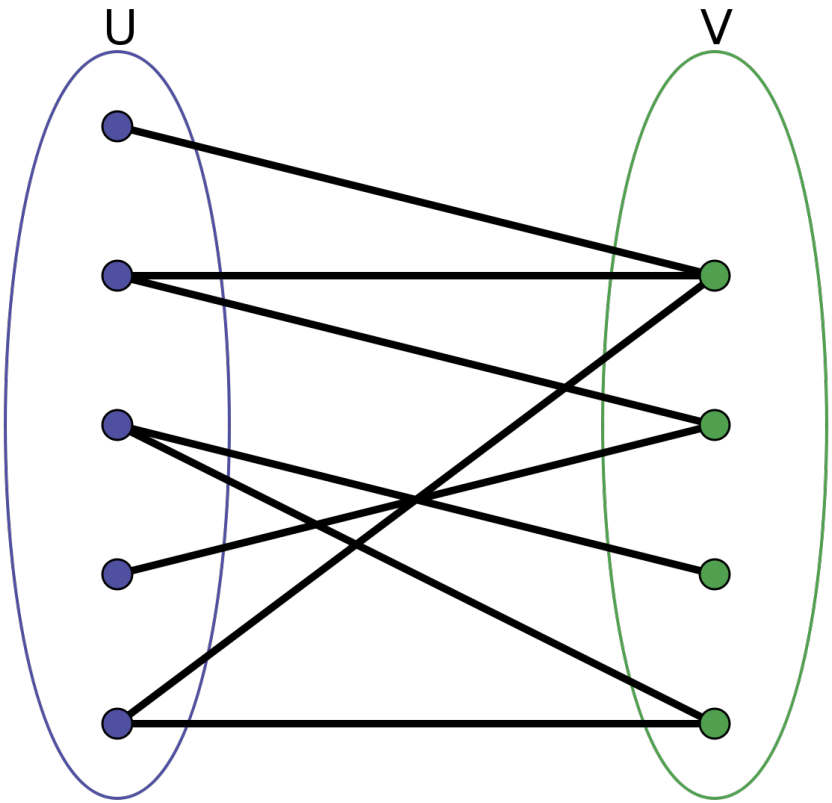
If every vertex is reachable from every other vertex

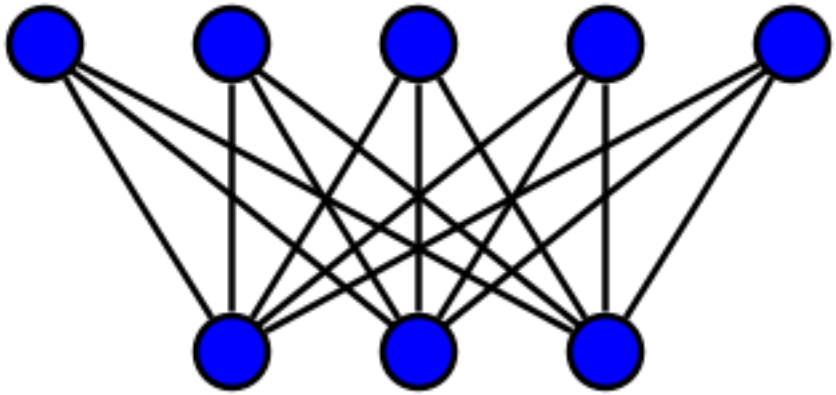
Strongly connected Components in a Digraph



If every vertex is reachable from every other vertex within the same component

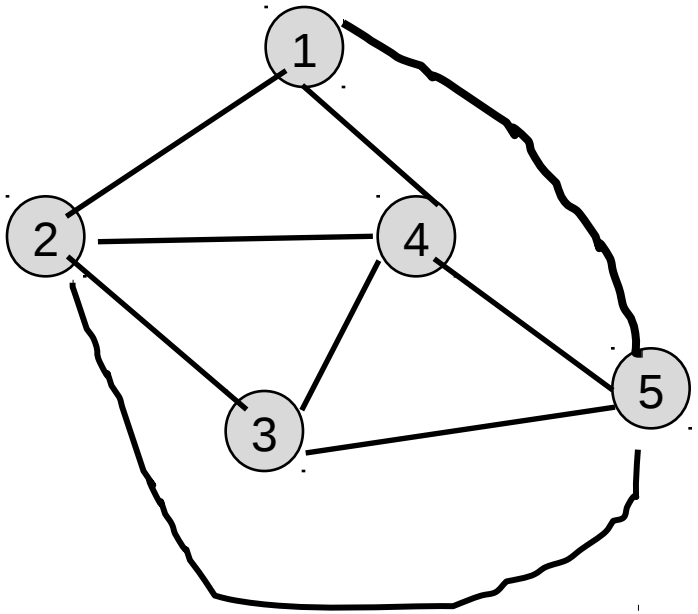
Bipartite Graph





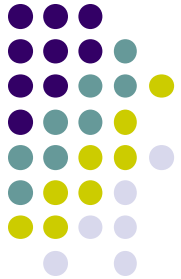
- U and V are disjoint sets of vertexes
- Every vertex in U connects to a vertex in V, and viceversa.

Complete graph



V(V-1)/2 edges

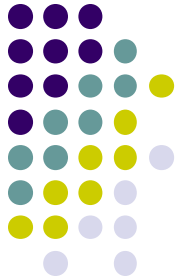
Trees and Graphs



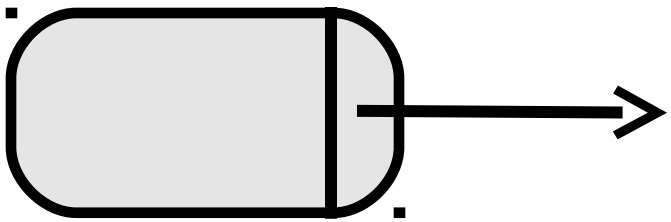
- Tree: a undirected graph that is:
  - Connected
  - Acyclic

- *n.b.* Any two vertices are connected by exactly one simple path.
- *n.b.* All vertices are connected.

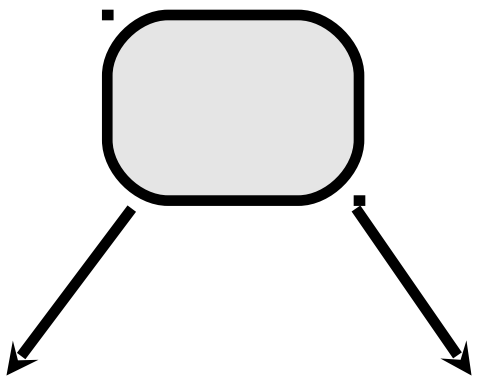
Representing graph vertices



- Linked list nodes:

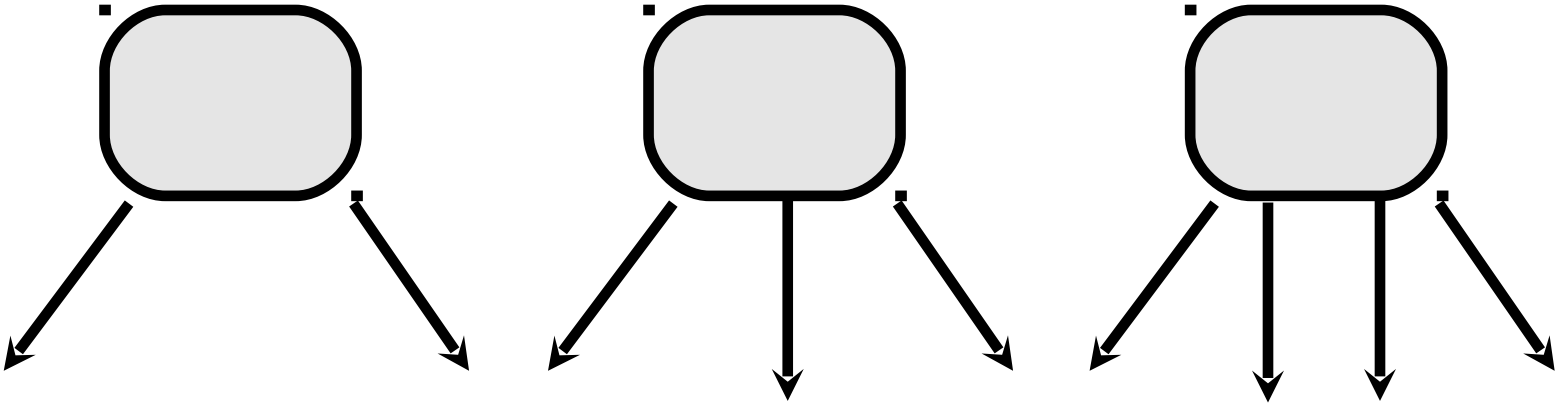


- Binary tree nodes:



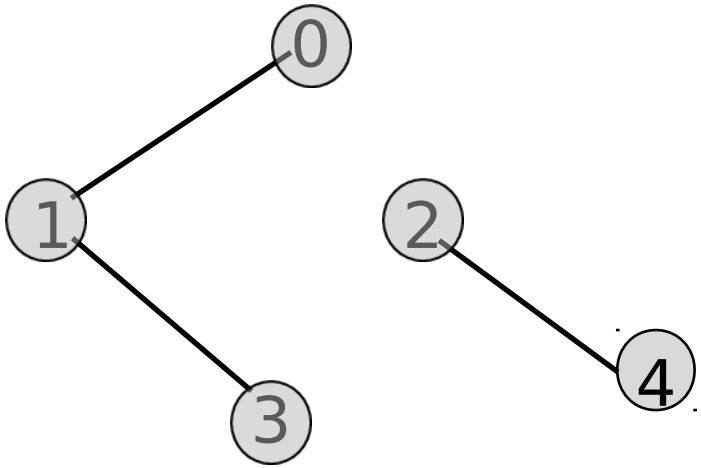
Vertices

- 2-3-4 tree nodes



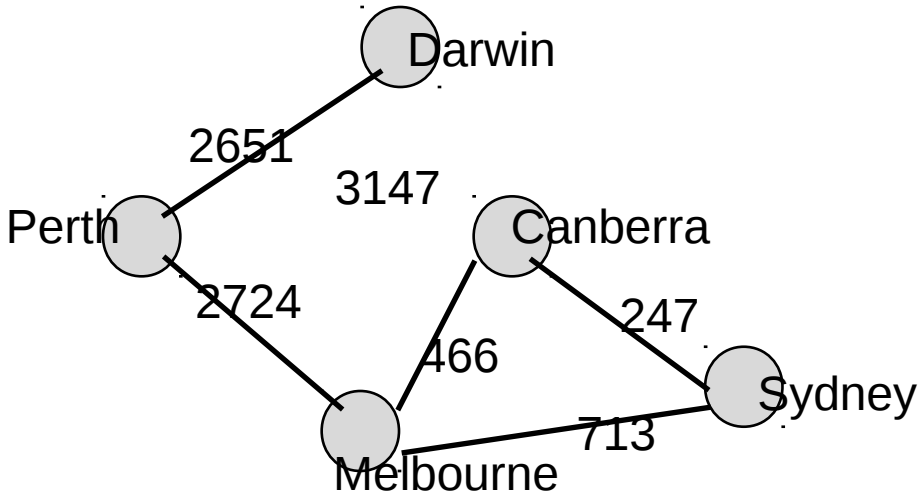
- But what if the number of links is potentially large (up to V-1)?

Array representation



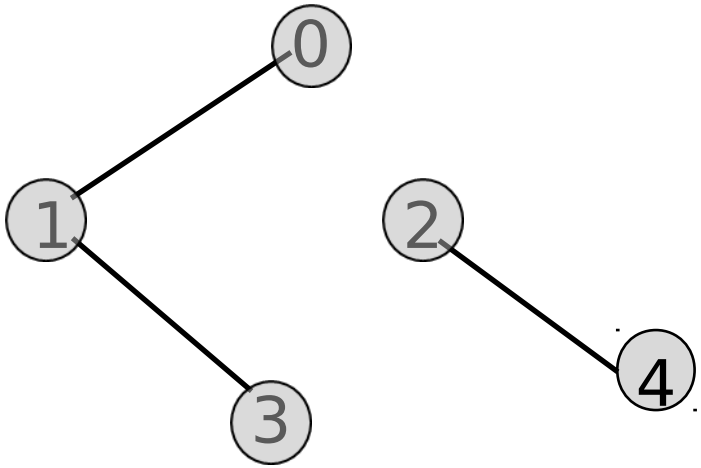
A	0	1	2	3	4
0	0	1	0	0	0
1	1	0	0	1	0
2	0	0	0	0	1
3	0	1	0	0	0
4	0	0	1	0	0

Weighted undirected graph



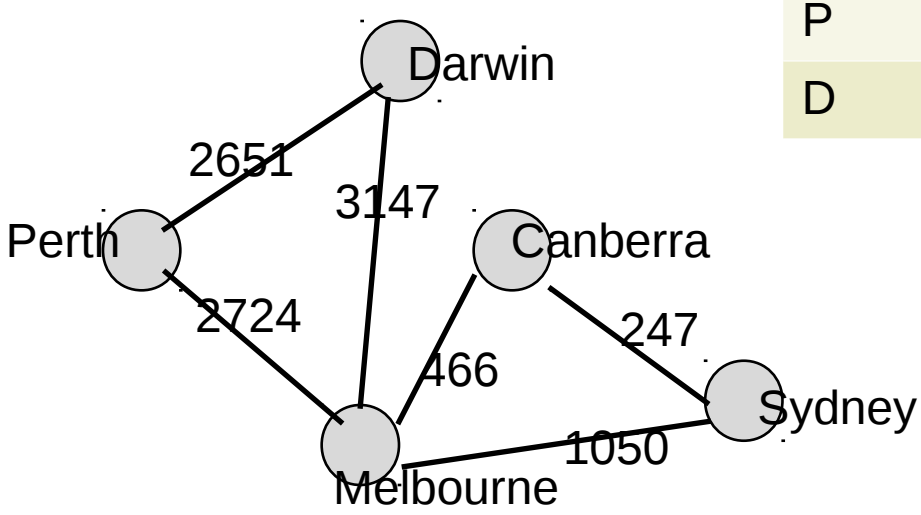
713

Array representation of weighted undirected graph



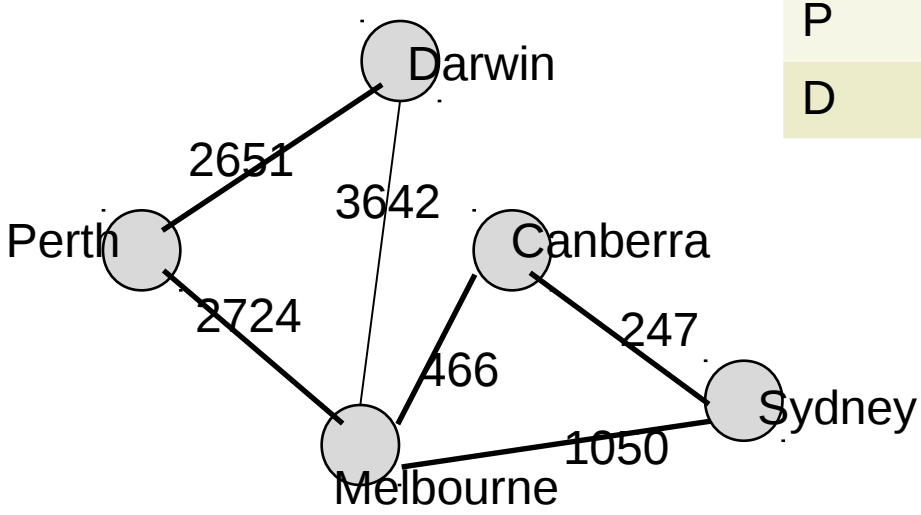
A	0	1	2	3	4
0		25			
1	25			40	
2					10
3		40			
4			10		

	M	C	S	P	D
M	0	466	1050	2724	3642
C	466	0	247		
S	1050	247	0		
P	2724			0	2651
D	3642			2651	0



713

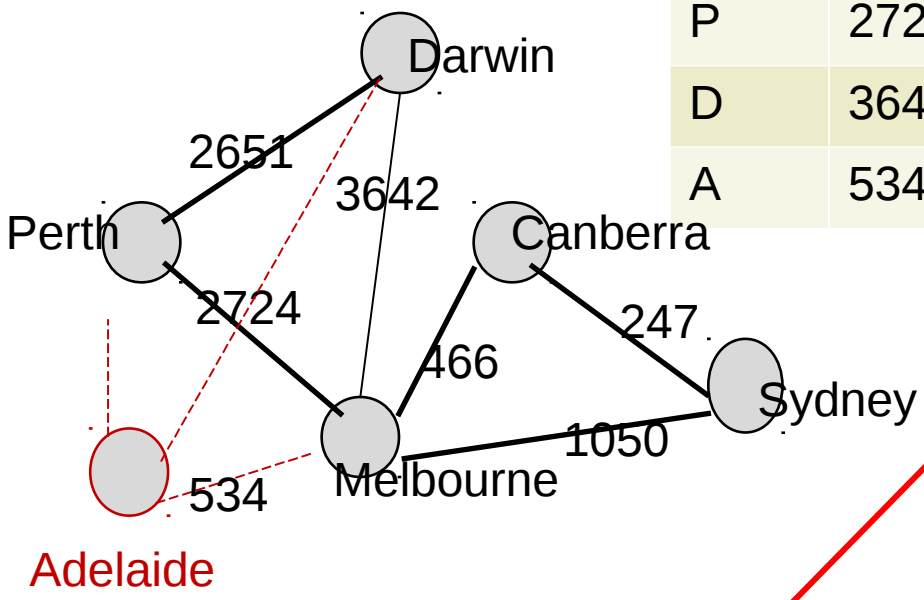
	M	C	S	P	D
M	0	466	1050	2724	3642
C	466	0	247		
S	1050	247	0		
P	2724			0	2651
D	3642			2651	0



Canberra to Darwin?

466+3642=4108

	M	C	S	P	D	A
M	0	466	1050	2724	3642	534
C	466	0	247			1160
S	1050	247	0			1163
P	2724			0	2651	2134
D	3642			2651	0	2620
A	534	1160	1163	2134	2620	0



Canberra to Darwin?

Still 466+3642?

Or 466+534+2724?

Shortest route between Melbourne and Dubbo?

Considerations:

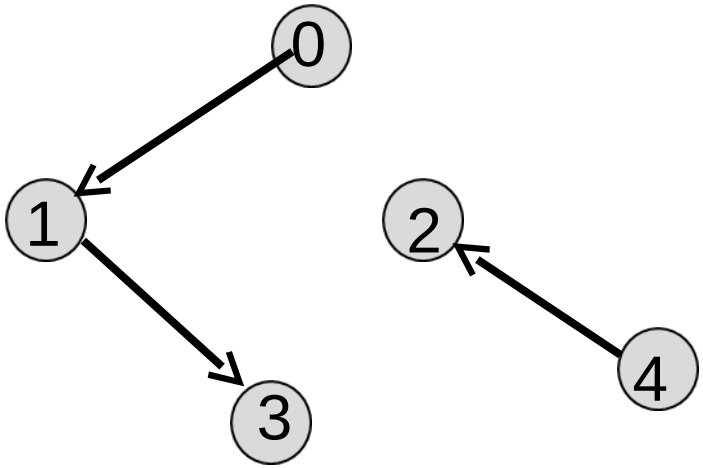
- Distance
- Kind of road
- Traffic points

Shortest route between Melbourne and Dubbo?

Dubbo

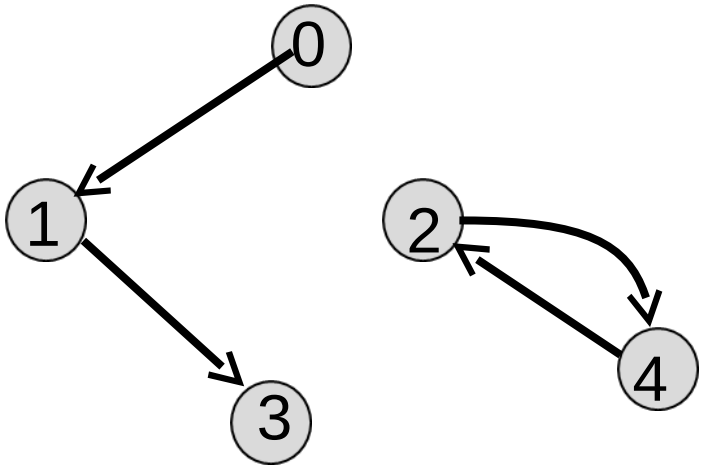
Dubbo

Array representation: weighted directed graph



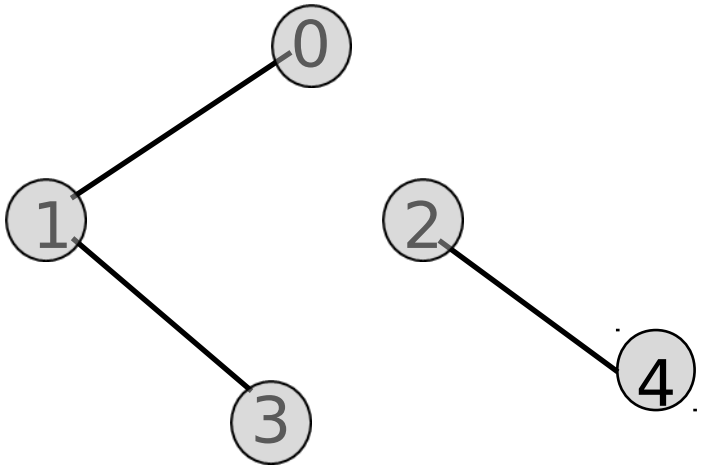
A	0	1	2	3	4
0		25			
1				40	
2					
3					
4			10		

Array representation: weighted directed graph



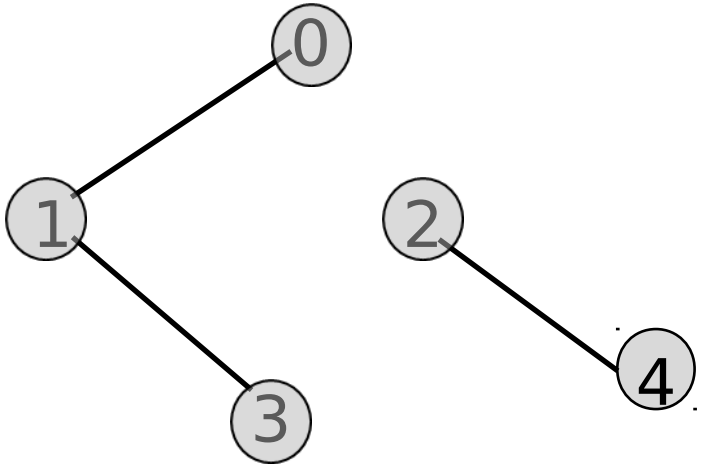
A	0	1	2	3	4
0		25			
1				40	
2					15
3					
4			10		

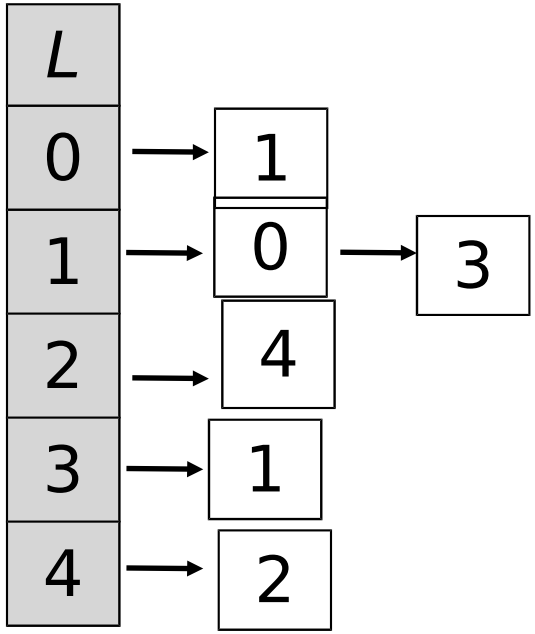
Array representation: initialization



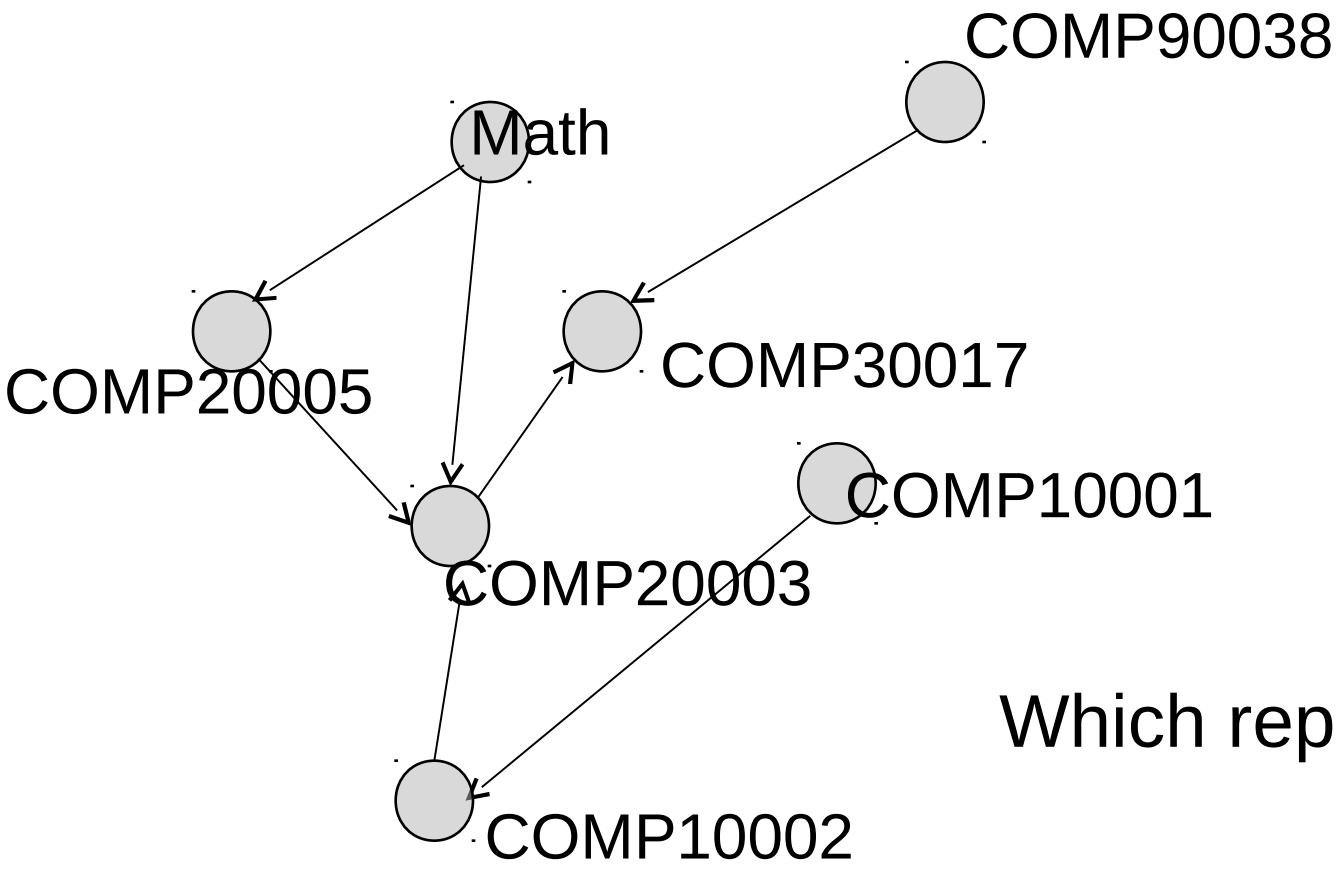
A	0	1	2	3	4
0	∞	25	∞	∞	∞
1	25	∞	∞	40	∞
2	∞	∞	∞	∞	10
3	∞	40	∞	∞	∞
4	∞	∞	10	∞	∞

Adjacency list representation



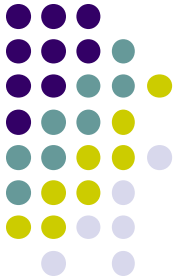


Directed graph: Subject prerequisites



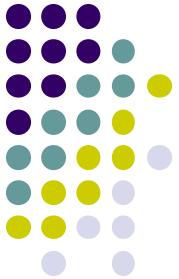
Which representation?

Size of matrix and list



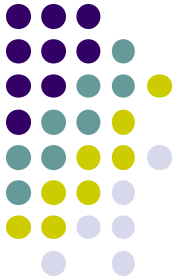
- Size of representation in terms of
  - |V| number of vertices
  - |E| number of edges
- Matrix
  - O(??)
- Adjacency list
  - O(??)

Size of matrix and list



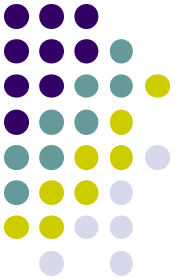
- Matrix
  - $O(|V|^2)$
- Adjacency list
  - $O(|V|+|E|)$
- Dense graph, lots of edges, use matrix representation.
- Sparse graph, use list.

Some interesting graph path problems



- Reachability.
- Single shortest path.
- Single source shortest path.
- All pairs shortest paths.
- Travelling Salesman Problem.

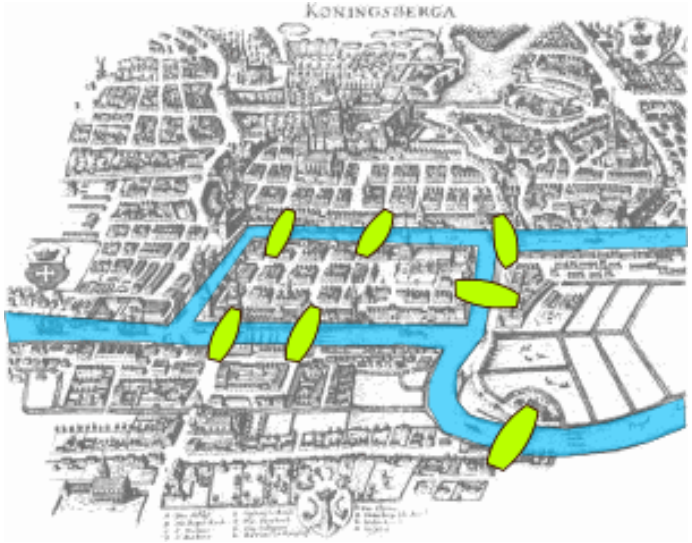
Other interesting graph problems



- Minimum Spanning Tree
- Topological sort
- Map coloring
- Matching

History

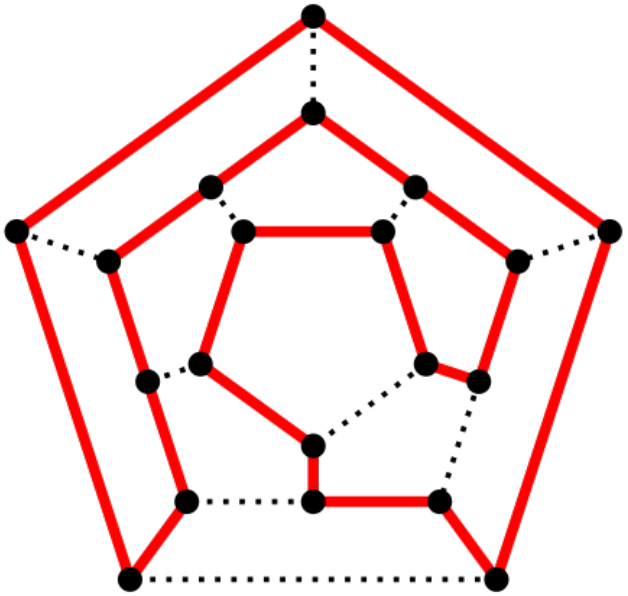
- Graph theory started with Euler (1736) who was asked to find a nice path across the seven Königsberg bridges



- The (Eulerian) path should cross over each of the seven bridges exactly once

History

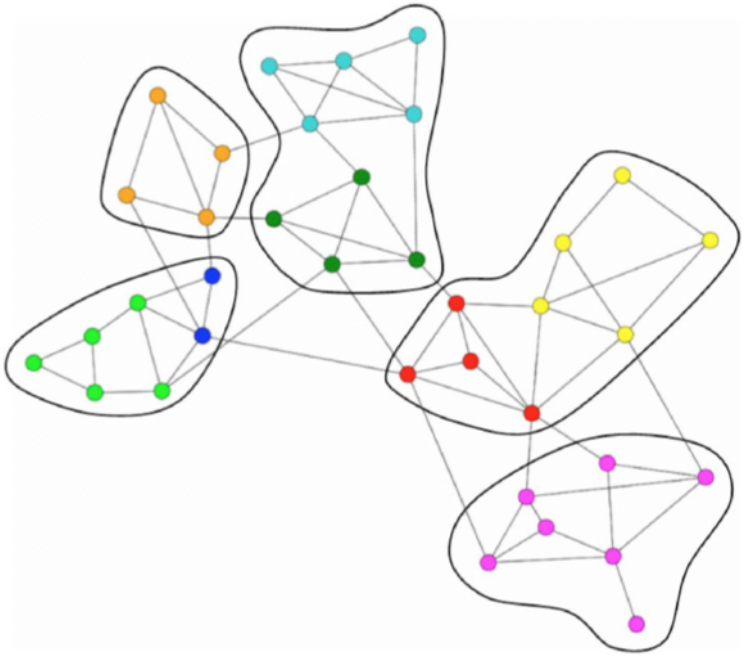
- Another early precursor was Sir William Rowan Hamilton (1805-1865)



- In 1859 he developed a toy based on finding a path visiting all cities in a graph exactly once and sold it to a toy maker in Dublin. It never was a big success.

Applications

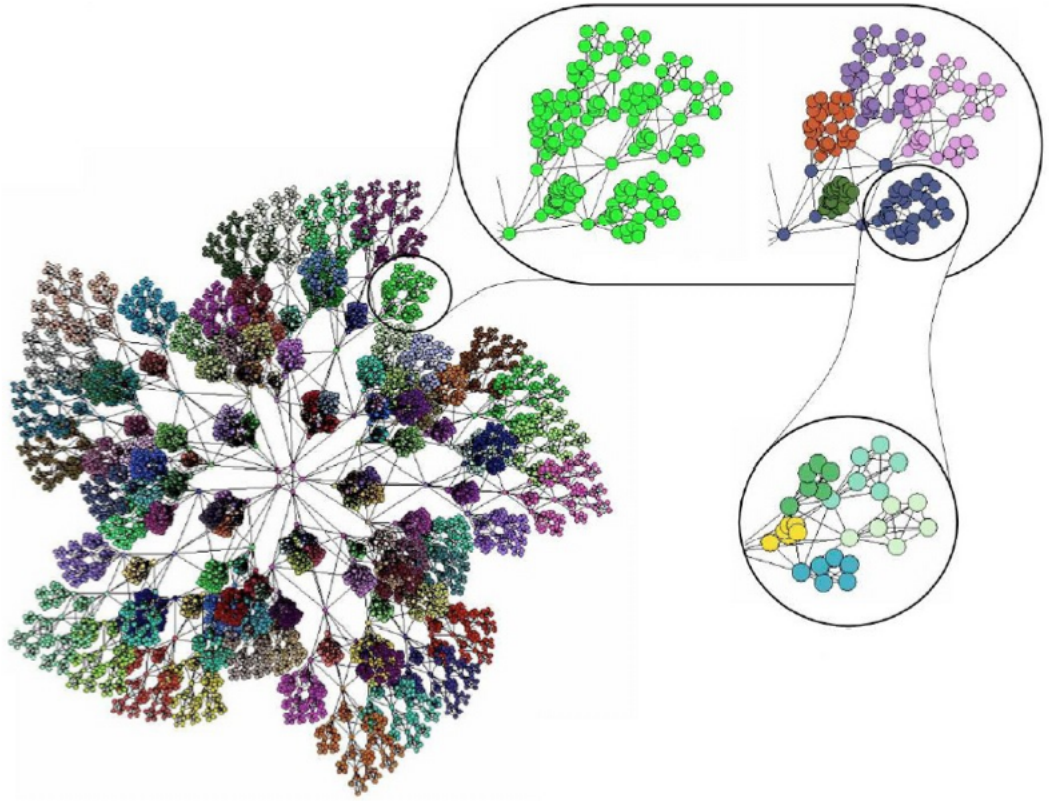
- But now graph theory is used for finding communities in networks



- where we want to detect hierarchies of substructures

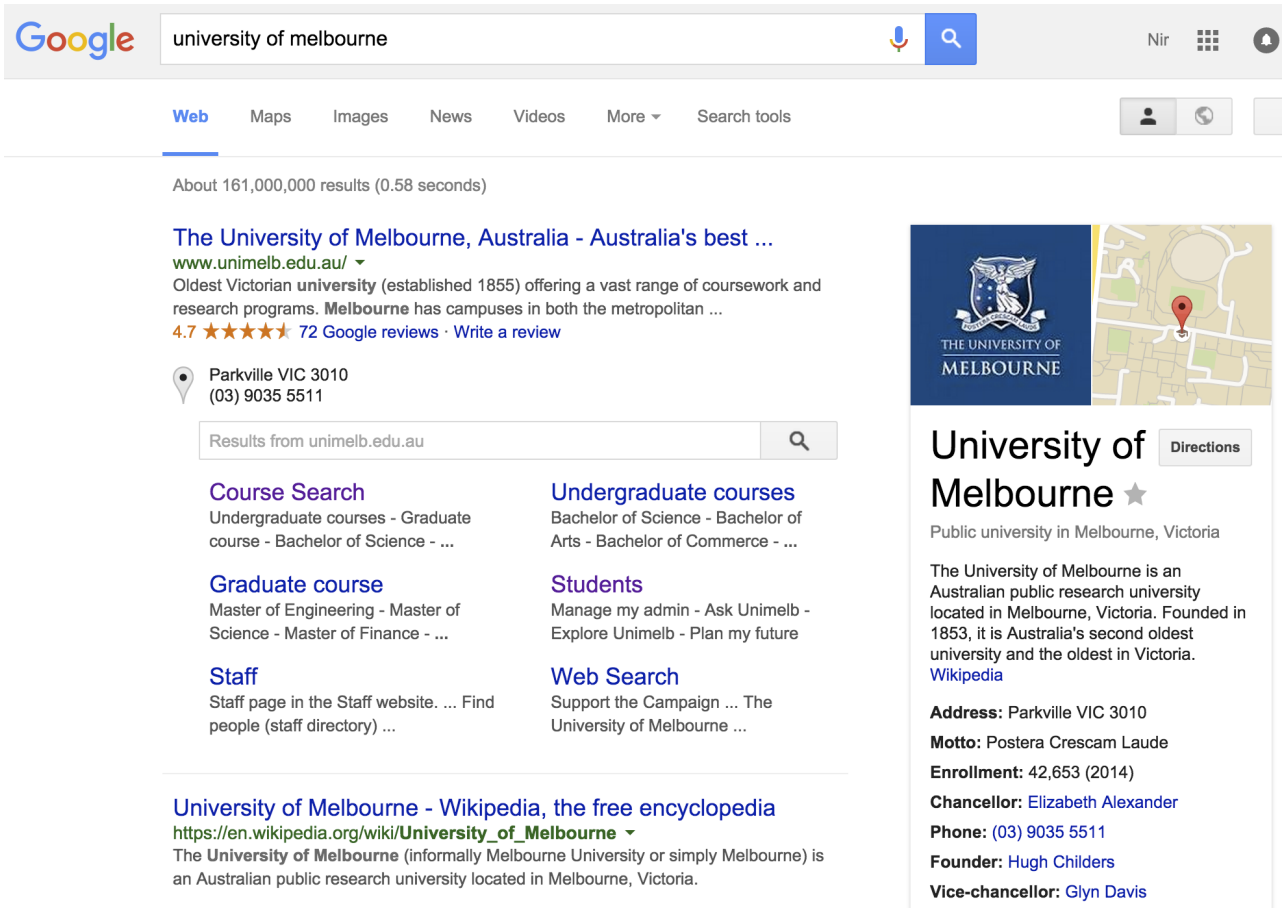
Applications

- and their sizes can become quite big ...



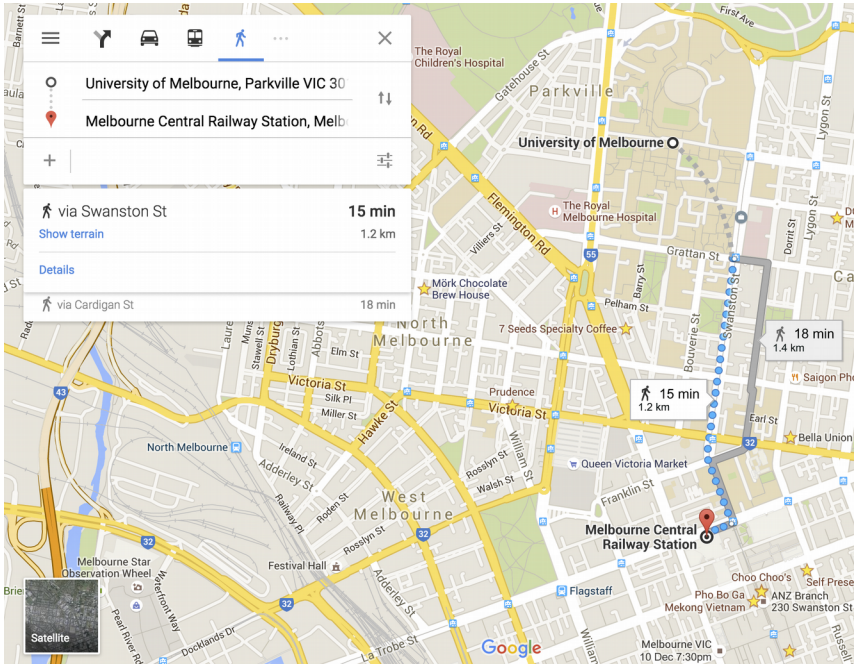
Applications

- It is also used for ranking (ordering) hyperlinks



Applications

- or by your GPS to find the shortest path home ...



- … and the list should go on for hours