Lecture 9
Graphs and trees

COS10003 Computer Logic and Essentials (Hawthorn)

SWINBURNE UNIVERSITY OF TECHNOLOGY

Semester 1 2021

Today

1. Foundation
2. Paths and cycles
3. Algorithms
4. Search
5. Trees
6. Applications

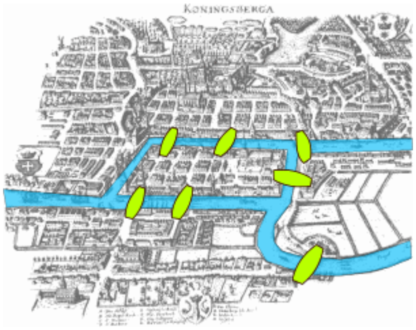
Defining graphs and trees

Exploring graphs with special properties

Algorithms for shortest paths

Let's start with a problem

In 1736 a town in Prussia had seven bridges as shown. How is it possible to walk around the city crossing each bridge exactly once?



`https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png`, CC SA 3.0

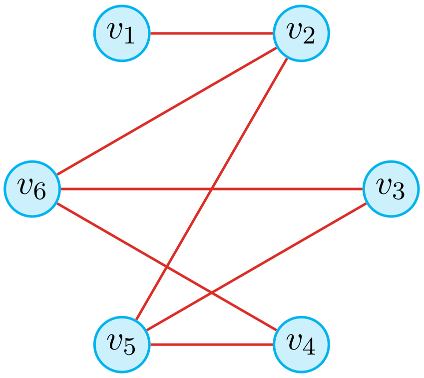
What is a graph?

A graph G consists of two sets:

- A set V whose elements are called vertices, points or nodes.
- A set E of unordered pairs of distinct vertices, called edges or links.

We denote a graph by G(V, E) when we want to emphasise the two parts of G.

Visualising

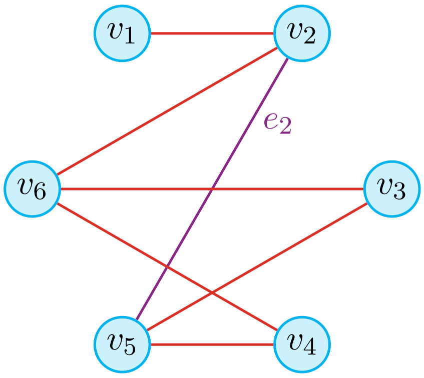


$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$
$E = \{\{v_1, v_2\}, \{v_2, v_5\},$
$\{v_2, v_6\}, \{v_3, v_5\}, \{v_3, v_6\},$
$\{v_4, v_5\}, \{v_4, v_6\}\}$

More definitions

- Vertices u and v are said to be adjacent if there is an edge e between them, that is (u, v).
- The edge e is said to be incident to u (and v).
- We can have multiple or parallel edges between the same vertices, however this then creates a multigraph.
- The degree of a vertex is the number of incident edges it has.

For you to do

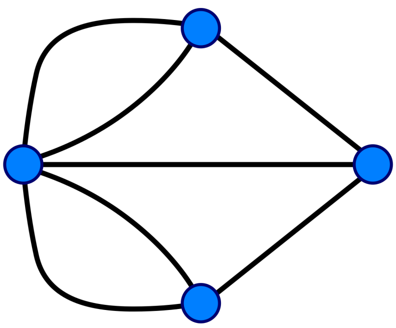


- The degree of v_4? **2**
- The degree of v_5?
- v_1 is adjacent to?
- v_3 is adjacent to?
- e_2 is incident to?

Revisiting our problem

In 1736 a town in Prussia had seven bridges. How is it possible to walk around the city crossing each bridge exactly once?

The problem becomes how to traverse every edge once. Can this be done?



`https://commons.wikimedia.org/wiki/File:Konigsburg_graph.svg`, CC SA 3.0

Before we continue

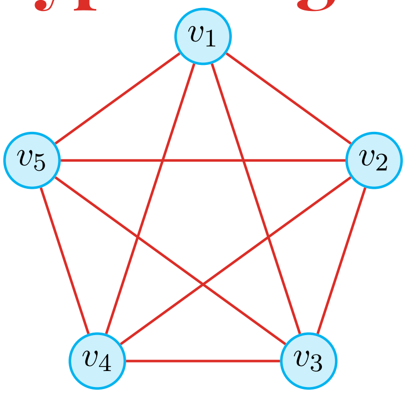
Graphs are an example of how data can be represented for computation purposes.

- Computers that are connected in some way in a network
- People who work together or are linked on a website
- Representation of possible paths to take for a robot or (autonomous) vehicle
- Allocation of resources such as timetabling or scheduling

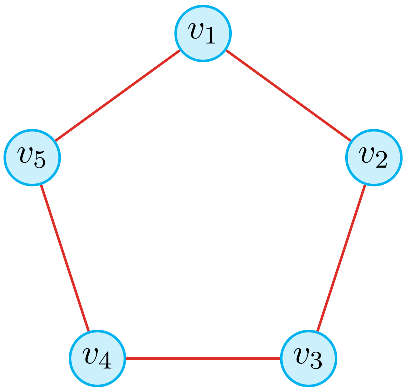
More on degree

The degree sum formula says that the sum of degrees in a graph G is twice the number of edges, that is, $\sum_{v\in V} \deg v = 2|E|$.

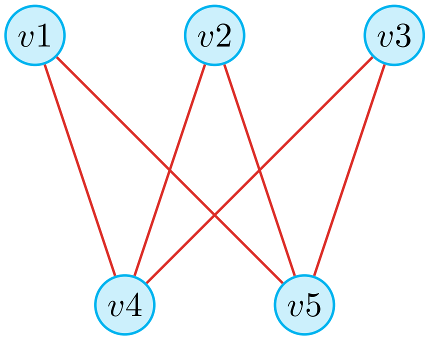
Types of graphs



K_n: the complete graph with n vertices (K_5 shown)



C_n: the cycle on n vertices (C_5 shown)



$K_{m,n}$: the complete bipartite graph with sets of m and n vertices ($K_{3,2}$ shown)

Bipartite graphs

A bipartite graph is a graph for which it is possible to divide the vertices into two disjoint sets such that there are no edges between any two vertices in the same set.

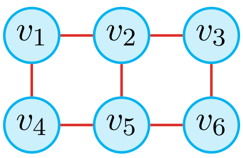
Subgraphs

A subgraph $G'(V', E')$ of a graph $G(V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$, and the vertices incident to any edges in E' are in V'.

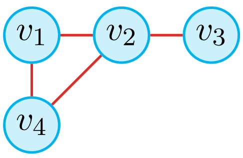
For you to do

Which graphs are subgraphs of G_1?

G_1:

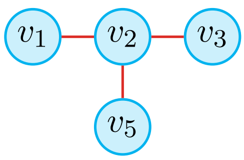


G_2:

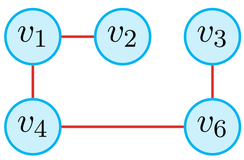


No

G_3:



G_4:



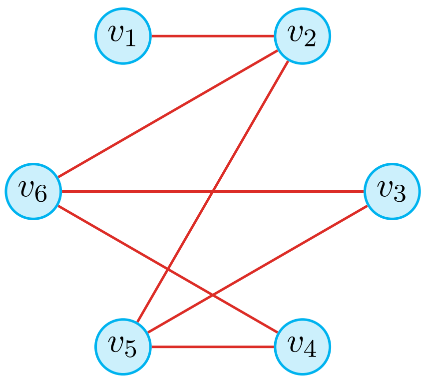
Isomorphic graphs

Two graphs are isomorphic if they are the same except for the naming of the vertices.

Look for:

- same number of vertices and edges
- same number of vertices with certain degrees
- same cycles

Paths

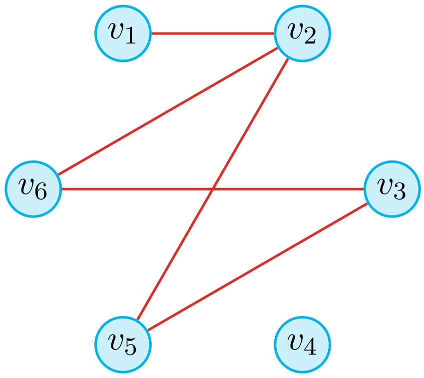


A path through a graph is an alternating sequence of $n + 1$ vertices and n edges, so that edge e_i is incident to vertices v_{i-1} and v_i.

What is a path from v_1 to v_3? (v_1, $\{v_1, v_2\}$, v_2, $\{v_2, v_6\}$, v_6, $\{v_6, v_3\}$, v_3)
This is a path of length 3; count the number of edges.

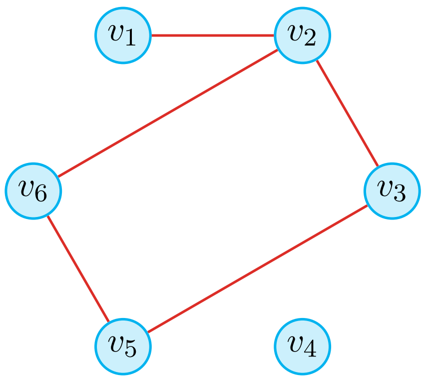
Connected graphs

A graph G is connected if for any vertices v and w in G, there is a path from v to w.



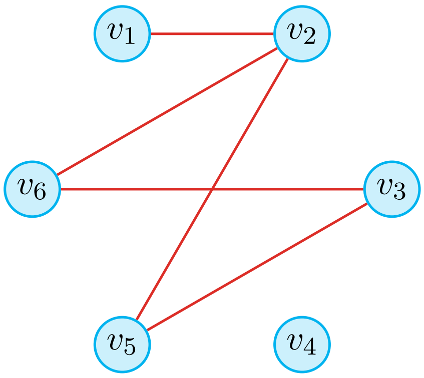
Cycles

A cycle is a path of non-zero length from vertex v to itself, with no repeated edges. A simple cycle has no repeated vertices apart from v at the start and end.



There is a cycle $(v_2, v_3, v_5, v_6, v_2)$.

For you to do

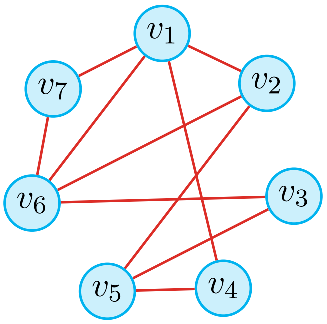


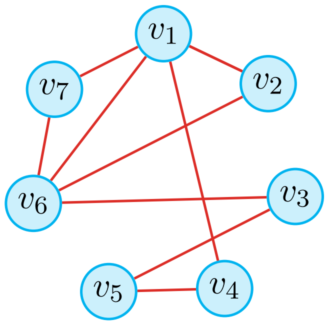
- Find a path between v_1 and v_5. (v_1, $\{v_1, v_2\}$, v_2, $\{v_2, v_5\}$, v_5)
- What is the length of the path?
- Find a path between v_2 and v_4.
- Find the longest simple path between v_1 and v_5?
- Find a cycle from v_3.

Eulerian cycle

An Eulerian cycle is a cycle that uses all the edges in the graph exactly once and starts and ends at the same vertex.
For this to work, the graph must be connected and each vertex must have an even degree.

Examples

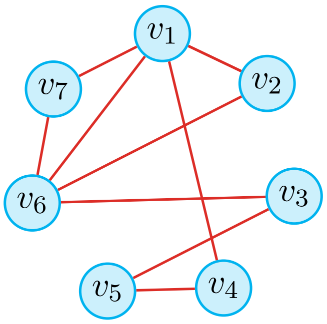


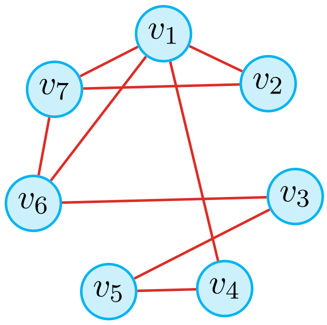


Hamiltonian cycles

A Hamitonian cycle is a cycle where each vertex is visited exactly once, apart from the starting and ending vertex which is the same.
This is considered to be a very difficult problem to solve; there are no definite tips on properties to look for.

Examples





Lecture 9

Graphs and trees

COS10003 Computer Logic and Essentials (Hawthorn)

Semester 1 2021

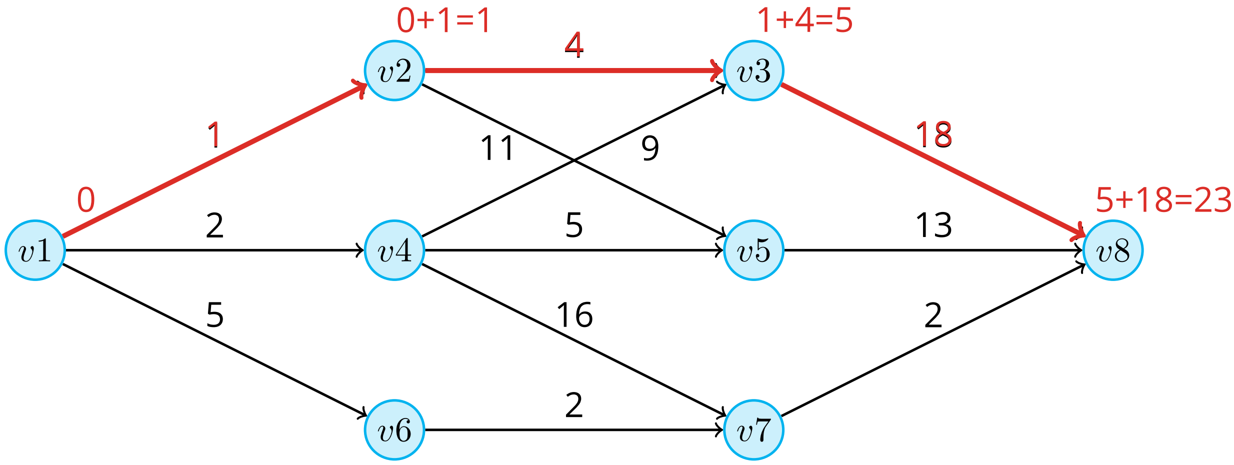
Directed graphs

- In many case we want to work with edges that have a direction (a source vertex and a sink vertex).
- These are often weighted as well, signifying some sort of cost to traverse those links.
- This weight could be distance, cost, or time for example.

Shortest path

Given any two nodes, find the path joining them of minimum weight.
Assume all weights are positive.

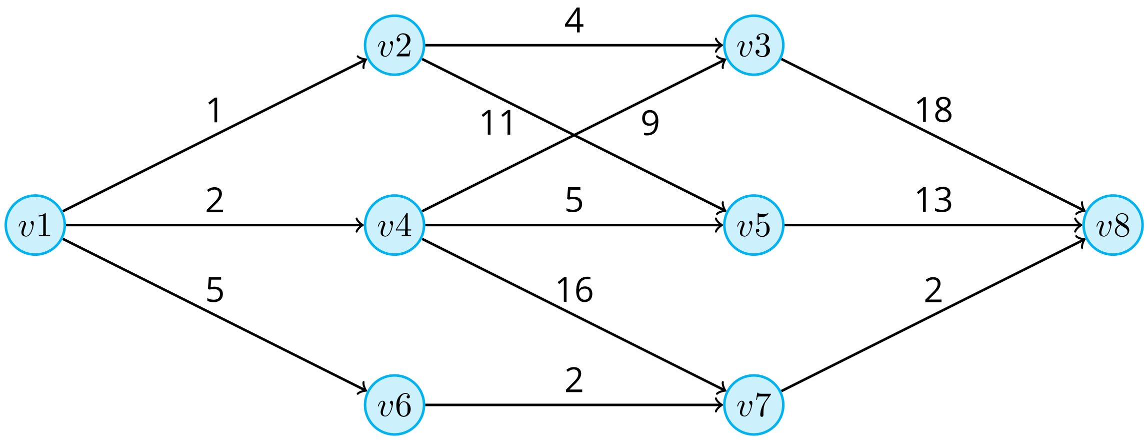
A greedy approach



Dijkstra’s algorithm

1. Mark all nodes unvisited. Create a set of all the unvisited nodes called the unvisited set.
2. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes. Set the initial node as current.
3. For the current node, consider all of its unvisited neighbours and calculate their tentative distances through the current node. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. Otherwise, keep the current value.
4. When we are done considering all of the unvisited neighbours of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new ”current node”, and go back to step 3.

Dijkstra in action

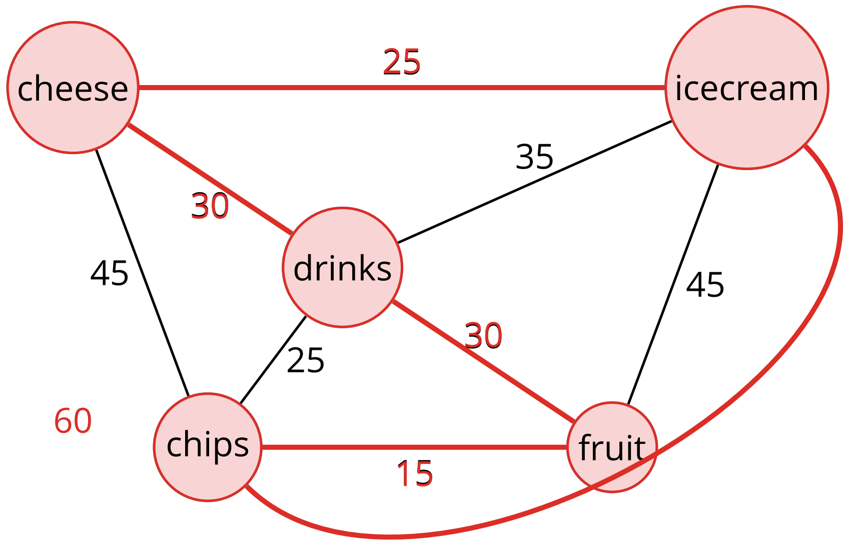


Foundation Paths and cycles Algorithms Search Trees Applications Next

Another problem

How can we visit each vertex exactly once and return to the starting point with minimum total weight?
This is known as the travelling salesman problem.

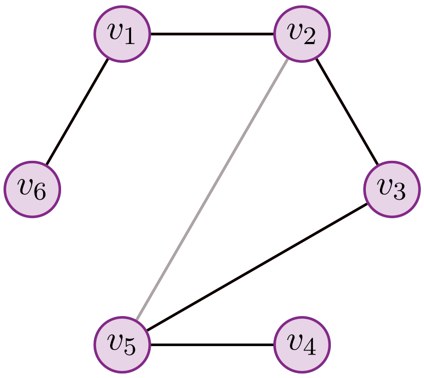
Travelling Salesman Problem



Types of algorithms

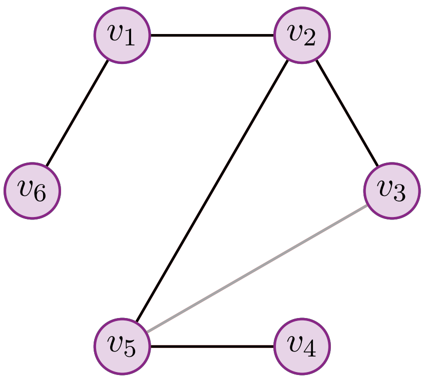
- Exact algorithms produce the correct solution, however tend to work only for small problems due to prohibitive runtime.
  - Enumerate every path (TSP)
  - Dijkstra's algorithm
- Heuristics have better runtimes and will produce a solution, but possibly not the correct solution.
  - Greedy shortest path
  - Nearest neighbour

Depth-first search



- start at v_1, see v_2 and v_6
- move to v_2, see v_3 and v_5
- move to v_3, see v_5
- move to v_5, see v_4
- move to v_4, no more nodes to visit
- move back to v_1 and follow v_6

Breadth-first search

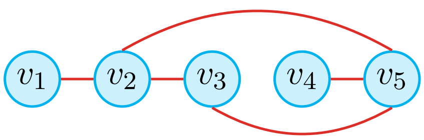


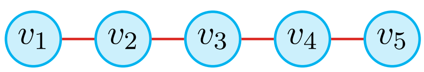
- start at v_1, see v_2 and v_6
- move to v_2, see v_3 and v_5
- see v_6
- move to v_3
- move to v_5, see v_4
- move to v_4, no more nodes to visit

What is a tree?

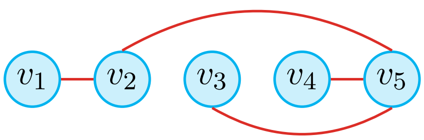
- A tree is a graph T where there is a unique simple path between all pairs of vertices u and v in T.
- Another way of saying this is that a tree with n vertices has $n-1$ edges and is connected.
- Trees are acyclic, that is, they have no cycles.
- Trees are commonly used in computer science for representing the organisation of data or for decision problems.

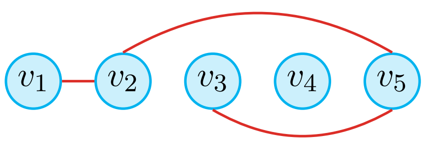
Which of these are trees?





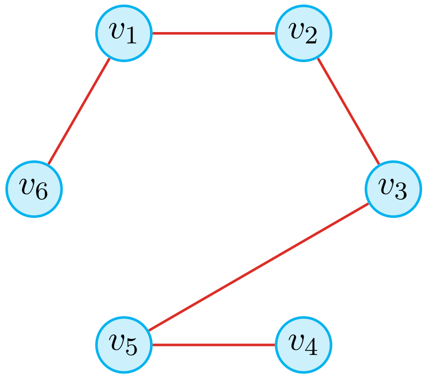
No





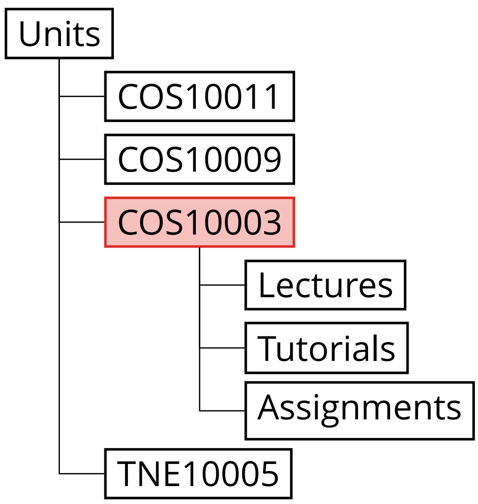
Spanning tree

A spanning tree is a subset of a graph with $n-1$ edges.



Rooted tree

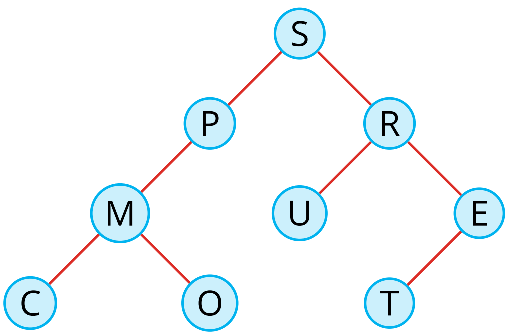
A rooted tree is a tree with one element named as the root element. The remaining elements are in a hierarchy.



Tree traversal

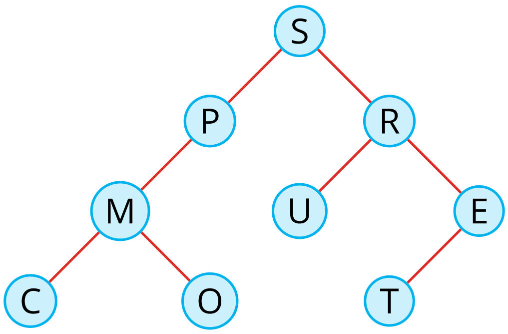
- Preorder
- In-order
- Postorder

Pre-order



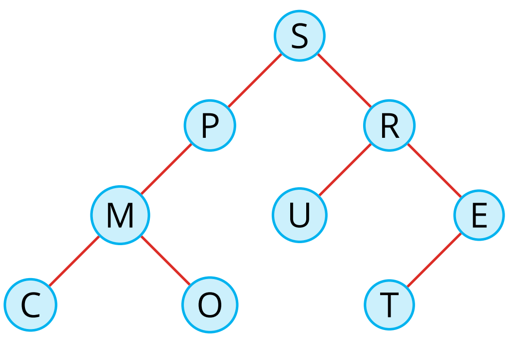

```
pre-order(node):
   if the current node is empty
      return
   else
      print current node
      pre-order(left node)
      pre-order(right node)
```

In order




```
in-order(node):
   if the current node is empty
      return
   else
      in-order(left node)
      print current node
      in-order(right node)
```

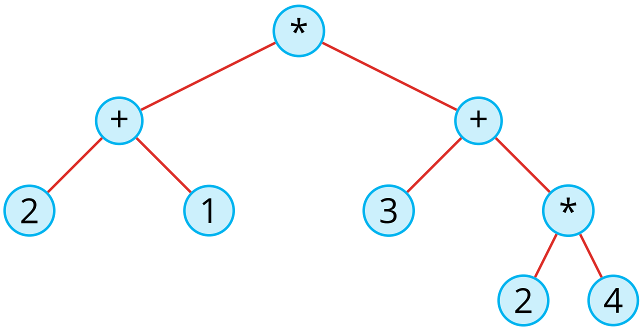

Post-order




```
post-order(node):
   if the current node is empty
      return
   else
      post-order(left node)
      post-order(right node)
      print current node
```

For you to do

What is the result of traversing this tree using post-order traversal?

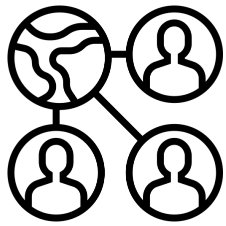


Some applications

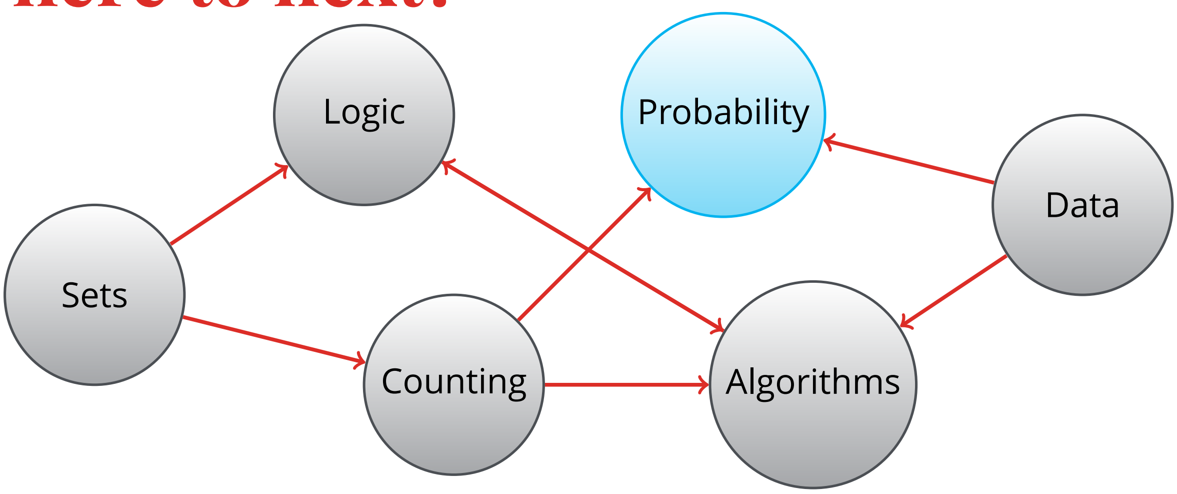
- Path planning and optimisation problems, e.g., for robotics, non-autonomous vehicles
- Puzzles and games
- Graph databases: more obhect-oriented than relational
- Medicine, biology, neuroscience: disease contagion, DNA, brain wiring

Reflecting

- What are the main components of graphs?
- What are graphs useful for?
- How can you find the shortest path between two nodes?
- What are the specific characteristics of a tree?



Where to next?



In which we look at probability for computing.

Lecture 9
Graphs and trees

COS10003 Computer Logic and Essentials (Hawthorn)

Semester 1 2021

Questions I still have

Topics I need to review