# COMP20007 Design of Algorithms: Week 3

- Designing an intersection
- Graph representation
- An algorithm
- DFS
- complexity

Review

- Minimal spanning tree problem
- Representing the problem using known ADTs
  - adjacency list, adjacency matrix
- Algorithm for identifying the minimal set of connections

  - Two algorithms, led to ideas of cyclicity
    and connectedness
- Required operations
  - add_node: Graph × int → Graph
  - add_edge: Graph × int × int → Graph
  - Is_cyclic: Graph → Bool
- Formal definition
  - G = (V, E), V=vertex set, E=edge set
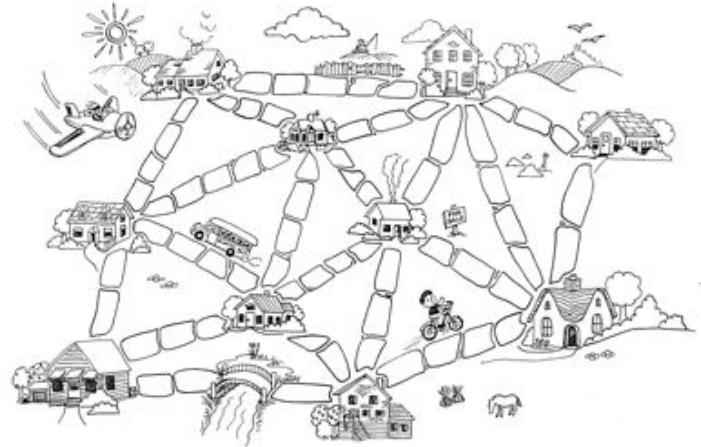
Topological sort

- Motivation
- Directed graphs
  - Sources and sinks
- Directed acyclic graphs (DAGs)
- Source removal algorithm
- Complexity?
- Critical path algorithm
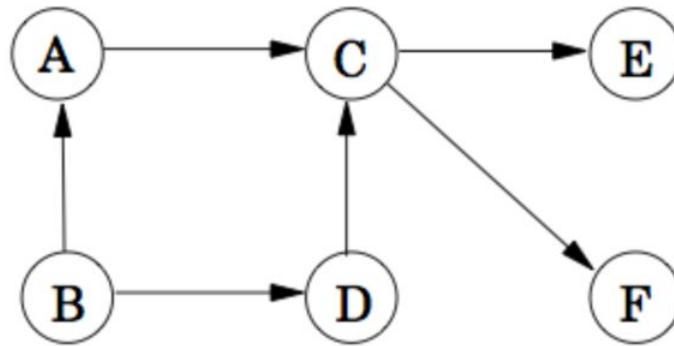
This week's labs and tutes

Implementation and analysis of median finding (k-smallest element of a list)

Value of kth-smallest item is d

Choose a pivot p and split into L and R.
Look at the size of L:
  ≥ k, then d is in L
  = k − 1, then p is d
  < k − 1, d in R:
      k − (|L| + 1)th-smallest in R

Either way, same problem,
but smaller input.

Next steps...

- Student rep(s)
- Read: [DPV 3.1, 3.2](#)
- Assignment 1 posted last Tuesday
- Week 2 tutorial solutions available
- News: [AlphaGo](#)