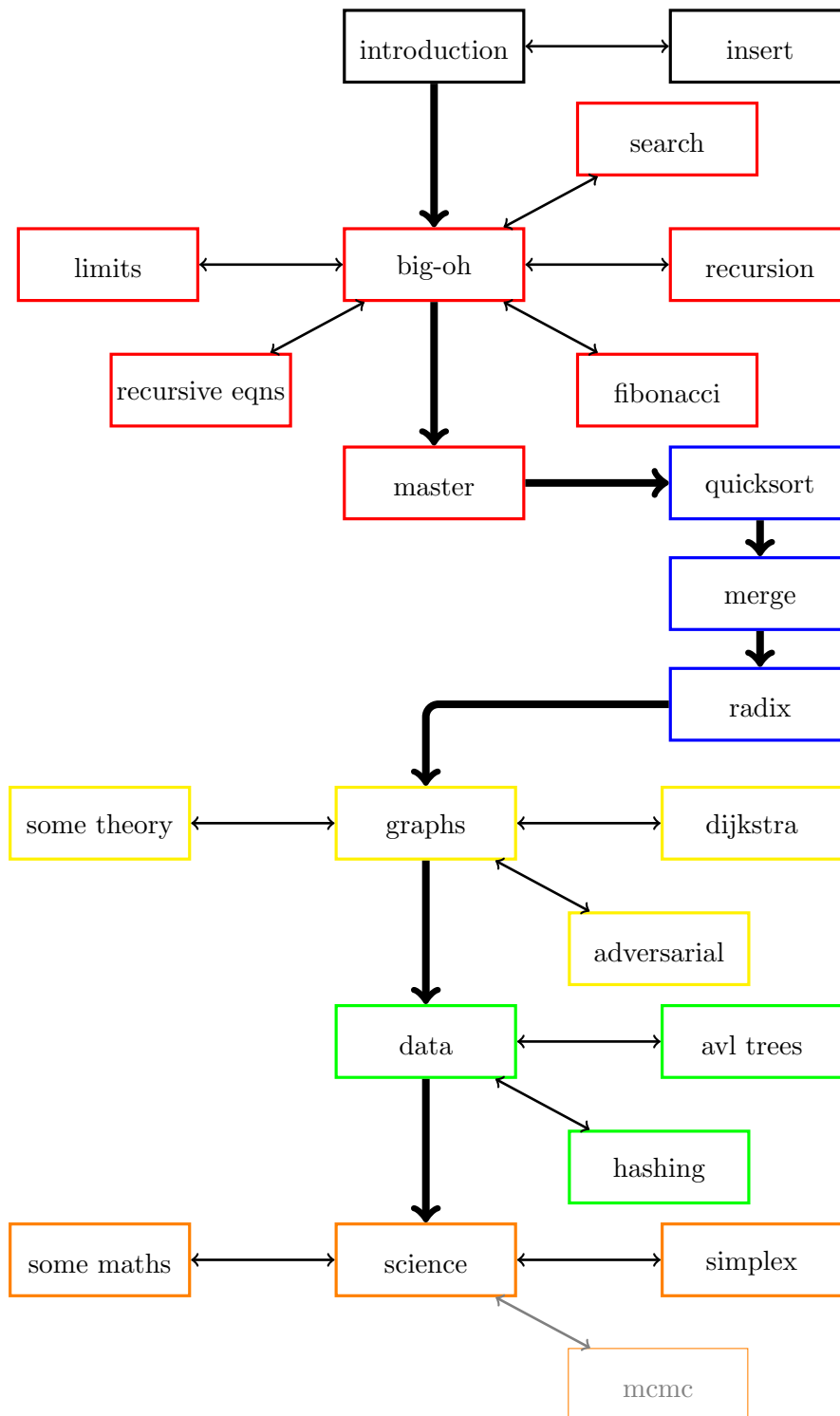


Course plan



Key to the plan

The nodes don't represent equal amounts of time, for example, the fibonacci sequence node represents about half a lecture, whereas the big-oh node might be two lectures.

Positions and colors

- CENTRE: main topics
- LEFT: background theory
- RIGHT: examples and applications
- RED: algorithmic complexity
- BLUE: sorting
- YELLOW: graph theory
- GREEN: data structures
- ORANGE: scientific computing

Nodes

- (a) **introduction:** What is an algorithm.
- (b) **insert:** Insert sort, an example algorithm.
- (c) **big-oh** Algorithmic complexity.
- (d) **limits** Limits in the mathematical sense.
- (e) **recursive eqns** How to solve recursive equations
- (f) **search:** Big-oh for binary and linear.
- (g) **recursion:** Recursion and recursive algorithms.
- (h) **fibonacci:** The interesting example of the fibonacci sequence.
- (i) **master:** The master theorem.
- (j) **quicksort:** Quicksort.
- (k) **merge sort:** Merge sort.
- (l) **radix sort:** Radix sort.
- (m) **graphs:** Introduction to graph theory.
- (n) **some theory:** Some definitions and the Euler theorem.
- (o) **dijkstra:** Dijkstra's algorithm.

- (p) **adversarial**: Adversarial search algorithms.
- (q) **data**: Data structures, linked lists and binary trees.
- (r) **avl trees**: Balanced trees and the AVL rotations.
- (s) **hashing**: Hash functions and hashing data.
- (t) **science**: Scientific computing.
- (u) **some math**: Revising a bit of calculus.
- (v) **simplex**: The Nelder Meade algorithm.
- (w) **mcmc**: Markov chain Monte Carlo

The last section is aspirational, it is likely there will not be time to discuss them and they will not be examined.