A force directed algorithm to draw undirected graphs:

- W. Tutte's algorithm was one of the very first force directed algorithms and can be used to draw undirected graphs. It worked by taking a set amount of nodes and situating them around the radius of a circle. After every other node was placed somewhere randomly within the circle, then each of these nodes would be placed at the barycentre of every other node it was connected too. It was essentially a minimization algorithm. This algorithm would be stopped after a few iterations of nodes. This method can be analoguised to the edges of each node being an elastic band influenced by their distance-squared. It has a complexity of O(n^3) to O(n^1.5) depending on implementation.

  Hooke's law is a variation of Tutte's algorithm and replaces the elastic band analogy with a spring algorithm, essentially edges are influenced by an extered force equal to | actual-Length - natural-Length | . Also – nodes repell one another via Coulomb's law of repulsion.

To draw directed graphs, the Sugiyama method can be used. The steps, optimisation criteria, and algorithms, are as follows:

- The steps are:
- 1. Cycle Removal.
    - Cycle removal ensures an acyclic graph, with all edges pointing downward, by temporarily reversing edges.
    - /Heuristics:

      /Fast Heuristic - Arbitary Ordering o, delete (u,v) where o(u)>o(v)

      /Enhanced Greedy Heuristic - If there's a source/sink 'u', otherwise choouse 'u' with max(outdeg-indeg), delete and add all incident edges to final graph. Linear Time&Space
    - /Randomized Algorithm: O(mn)
- 2. Vertical Assignment.
    - Some dummy vertices can be added.
    - /**Longest Path**: Place all sinks in L1, other v placed in L(p+1), (p=longestPathToL1). +LinearTime, –Wide. (Minimizes(H)).
    - /**Coffman-Graham**: Lexicographical Ordering. Restrict max(#v) in Ls. (W–Importance>H)
    - /**Network Simplex**: Minimize #dummyVertices. (Compact Drawing)
- 3. Ordering Assignment.
    - NP-Complete.
    - Layer-by-Layer Sweep: (2 layers at a time, 1 fixed, min(crossings)).
    - Sorting: Adjacency-Exchange[Greedy-Switch] (O(†L†^2) time, Split[Quick-Sort] (same upper, O(L.log(L)) avg)
    - Barycentre Method: Linear Time. At worst O(√n) times optimal.
    - Median Method: Linear Time. At worst 3 times optimal.
    - Integer Programming Method: (optimal, (no polytime end guarantee) (good4 s-M digraphs)
- 4. Horizontal Assignment.
    - Reduce angle of bends at dummy vertices.
    - Priority Barycentre Method: (Priority=Degree, dummyVs=HighestPriority) (Enforce minimal distance between adjacent vertices, move v of lowest priority).

Concepts:

- Scale-Free Networks (Internet (19), Metabolic, Citation)
  - 1. Number of nodes is not fixed. Exponential Growth.
  - 2. Attachment is not uniform. Preferential Attachment.
  - Power-Law Distribution. High Clustering Coefficient.
    Ultra-small avg path length: O(log log n).
- Centrality Analysis
  - Local Measures: degree, indegree, outdegree.
  - Distance (Global) Measures: Bewteeness (low degree can be important) - proportion of shortest paths passing through Y from X to Z. Closeness - sum shortest paths to all v. Eccentricity - length of longest shortest path.
  - Feedback Measures: status, hub, authority, eigenvector, pagerank.
- (Cohesive subgroups; components, cores,) Cliques
  - Components: Strong/Weak, Cycllic/ , connected/isolated, cut-vertex/separation-pair.
  - (Cliques are complete subgraphs where all nodes are connected.)
    n-clique, where n dictates the maximum path length of members of a clique, allow relaxation of definition with its increase. n-clan extends n-clique, requiring the diameter of the clique to be smaller than or equal to n.
    k-core finds dense areas where every vertex is **adjacent** (degree) to at least k other vertices. k-plex is less relaxed and finds a set of vertices in which every vertex is adjacent to All except k of the other vertices
    n-clique and n-clan are about reachability (path length). k-core and k-plex are about degree.
- Structural Equivalence
  - From strict to relaxed: Structural equivalence, Automorphic equivalence, Regular equivalence.
  - Structurally equivalent nodes hold identical positions in the network. A reduced graph, or blockmodel, essentially combines similar nodes into one. ("If a node in block x is connected to a node in block y, then every node in block x is connected to every node in block y."). Nodes that are structurally equivalent are also automorphically equivalent.
  - Automorphically equivalent nodes do not have to be connected to exactly the same nodes, but to nodes that play analogous roles in the network.
  - Regularly equivalent nodes have ties to the same role, revealing social structures. Regular equivalences relaxes the definition further, no longer requiring degree, but only that you know one person in a class.

Tidier Drawing Algorithm (Binary tree – Linear Time):

- 1. Postorder traversal – Compute Hortizontal Displacement of children of V, for every V. [÷,x]
- 2. Preorder traversal – Compute 'x' by accumulating displacements on path from V to root.
- Left/Right Contours for Postorder traversal. Displacements accumulated.
- Drawing: layered, Planar, straight-line, downward, O(n^2) area.
  Isomorphic subtrees have congruent drawings up to a translation.

Planar Radial Drawing (Free Tree - Linear Time):

- For free trees. Root is 'Centre'.
- Guaranteer Planarity by defining convex subset of the Wedge (everything falling below tangent of vertex).

HV-Drawing (Binary Tree):

- ÷: Recursively HV-draw L & R subtrees. x: Do H/V combination. [Hight&width each ≤ n-1]
- Right-Heavy-HV-Drawing:
  - Uses only H-Combination, placing subtree with the Largest number of vertices to the Right of the other one.
  - Area: O(n x logn). Good, bad aspect.
  - simply and axially isomorphic subtrees have congreuent drawings up to a translation.
- (Better Aspect Ratio than R-Heavy: Use H/V Combos for Odd/Even Depths)

.
[dot]
.

Questions:

- Describe an algorithm to draw free trees with radial drawings and analysis time complexity.
- Explain a force directed algorithm to draw undirected graphs.
- Explain each steps of the Sugiyama method to draw directed graphs including optimisation criteria and algorithms to achieve the criteria.
- Describe an algorithm to draw a planar graph with straight-line edges in O(n^2) area.
- Explain the following concepts:
    - Centrality analysis
    - Scale-Free networks
    - Cliques
    - Structural equivalence

An algorithm to draw free trees with radial drawings, and its' time complexity is as follows:

- The Tider Drawing Algorithm graphs a binary tree in linear time and works by doing a posorder traversal to compute the horizontal displacement of the children of vertex v, for every vertex. This is then followed by a preorder traversal to compute the actualy x-coordinate of every vertex. The y-coordinate can be predefined by the depth of each vertex. This same algorithm can be adopted for free trees in linear time. Similarly, this same algorithm can also be wrapped around (from a normal x-y coordinate assignment) to a radial (theta-radius assignment) in linear time. Hence, by adopting the Tidier Drawing Algorithm, a radial drawing for free trees can be achieved in linear time.