1. Graphs, BFS, DFS, Topological Sort
   -- Graph representation (adjacency list vs adjacency matrix)
   -- Graph searching technique: BFS and DFS
   -- BFS to find the number of connected components in an undirected graph
   -- DFS: Tree edge, back edge, forward edge, cross edge
   -- In a DFS of an undirected graph G, every edge is either a tree edge or a back edge [Proof: Theorem 22.10]
   -- DFS to detect cycle: A directed graph G is acyclic if and only if a DFS of yields no back edges [Proof: Theorem 22.11]
   -- For DFS, the discovery and finishing times of vertices have parenthesis structure [Theorem 22.7].
   -- Topological sort using DFS

2. Huffman coding
   -- fixed-length encoding
   -- variable length encoding
   -- compression gain/improvement

3. Single source shortest path
   -- Case where BFS doesn't work?
   -- Dijkstra's algo and finding the shortest path
   -- Case where Dijkstra's algo doesn't work?
   -- Bellman-Ford algorithm and finding the shortest path
   -- Case where Bellman-Ford/any shortest path algo doesn't work?
   -- optimal substructure property

4. Dynamic Programming (DP)
   -- Basic differences of greedy, divide-and-conquer, and DP
   -- Floyd-Warshall Algorithm to find all-pairs shortest paths
   -- Finding the transitive closure of a directed graph

-- Longest common subsequence (LCS)
-- Optimal Substructure Property in LCS/Transitive closure/all-pairs-shortest-paths

5. Minimum Spanning tree
   -- What is spanning tree?
   -- Find spanning trees from a given graph
   -- What is minimum spanning tree
   -- Kruskal's & Prim's algo to find the minimum spanning tree

** Please make sure you know the running time of all the algorithms included in these topics and go through the relevant lectures/problems/examples/suggested-reference-readings/hw-problems**