Fundamental Techniques

- Greedy Method
  - philosophy (greedy choice, substructure property)
  - problems
    - Fractional knapsack (algorithm, runtime)
    - Task scheduling (algorithm, runtime)
- Divide and Conquer
  - philosophy (divide, recur, conquer)
  - problems
    - Merge sort (algorithm, runtime)
    - Integer multiplication (algorithm, runtime)
  - recurrence equations and master theorem
- Dynamic Programming
  - philosophy (subproblem optimality, subproblem overlap, bottom-up, table)
    - define subproblems
    - show subproblem optimality
    - express solution to large problem through solutions to smaller problems (recurrence formula)
    - implementation
  - problems
    - matrix chain multiplication (solution, algorithm, complexity)
    - 0-1 knapsack problem (solution, algorithm, complexity)

Graphs

- Definitions: graph, vertex, edge, directed, weighted, vertex degree, adjacent, incident, path, simple path, cycle, simple cycle, subgraph, spanning subgraph, connected, connected components, spanning trees, forest, biconnected graph (components), separation vertex and edge
- Properties
- Data structures
  - edge list, adjacency list, adjacency matrix
  - performance
- DFS
  - algorithm and complexity
  - properties
    - connected component of v by DFS(G, v)
    - spanning tree by discovery edges (other edges are back edges)
  - applications
    - path finding
    - cycle finding
    - connectedness
    - connected components
    - spanning tree (forest)
    - biconnected components
      - be able to find (using any method) separation vertices, separation edges, and biconnected components
- BFS
  - algorithm and complexity
  - properties
    - connected component of v by BFS(G, v)
    - spanning tree by discovery edges (other edges are cross edges)
    - layering the vertices of G with $L_0, L_1, L_2, \ldots$
  - applications
    - connected components (connectedness)
    - spanning tree (forest)
    - cycle finding
    - path with minimum number of edges
- Comparison of DFS and BFS

Directed Graphs

- Definitions: in-degree, out-degree, directed path, reachability, directed cycle, DAGs, strong connectivity
- Representation: *v* has incoming edges and outgoing edges
- Directed DFS (complexity)
    - strong connectivity algorithm (complexity)
- Transitive closure
    - definition
    - algorithm (Floyd-Warshall) and complexity
- DAGs and topological sorting
    - any topological sorting algorithm and complexity

Weighted Graphs

- Shortest path problem formulation
- Shortest path tree and Dijkstra's algorithm (algorithm, complexity, applicability [no neg. edges])
- Bellman-Ford algorithm (algorithm, complexity, applicability [neg. edges OK, no neg-cycles])
- Shortest path in DAGs and linear-time algorithm (algorithm that uses topological sorting, applicability [neg. edges OK])
- **No** all pairs shortest path
- Minimum Spanning Trees
    - Definitions
    - Prim-Jarnik's algorithm
        - algorithm
        - complexity
        - properties behind the correctness - partition property and cycle property
    - Kruskal's algorithm
        - algorithm
        - complexity
        - how it differs from Prim-Jarnik approach
        - data structures and implementation (find, union)
    - **No** Baruvka's algorithm

Maximum Flow

- Definitions: edge capacity, flow network, source, sink, flow, cut, flow over cut, capacity of a cut
- Maximum flow problem formulation
- Flow augmentation and augmenting path
- Ford-Fullkerson's and Edmonds-Karp algorithm
    - be able to apply/use
    - complexity
- Max-Flow and Min-Cut Theorem