Fundamental Techniques

- Greedy Method
  - Philosophy (greedy choice, substructure property)
  - Fractional knapsack (algorithm, complexity)
  - Task Scheduling (algorithm, complexity)
- Divide and Conquer
  - Philosophy (divide, recur, conquer)
  - Merge sort (algorithm, complexity)
  - **I won't ask you about integer multiplication problem**
  - Recurrence equations and Master Theorem
- Dynamic Programming
  - Philosophy (define subproblems, subproblem optimality, subproblem overlap, bottom-up, table)
  - Matrix Chain Multiplication (algorithm, complexity)
  - 0-1 Knapsack Problem (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 & performance: edge list, adjacency list, adjacency matrix
- DFS (algorithm, 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, 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) **I won't ask you to show the execution of directed DFS**
  - strong connectivity algorithm (algorithm, complexity)
- Transitive closure (definition, Floyd-Warshall algorithm, complexity)

- DAGs and topological sorting (any topological sorting algorithm on DAGs and its complexity)

Weighted Graphs
- Single source shortest path problem formulation, shortest path tree
    - Dijkstra's algorithm, complexity, and applicability [no neg. edges]
    - Bellman-Ford algorithm, complexity, applicability [neg. edges OK, no neg-cycles]
    - Shortest path in DAGs and linear-time algorithm, applicability [neg. edges OK]
- All pairs shortest path (algorithm, complexity)
- Minimum Spanning Trees (definitions)
    - Prim-Jarnik's algorithm, complexity
    - Kruskal's algorithm, complexity
    - **I won't ask you about 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
- Max-Flow and Min-Cut Theorem
- Ford-Fullkerson's algorithm, complexity
- Edmonds-Karp algorithm, complexity