Fundamental techniques

- The greedy method
  - philosophy (greedy choice, substructure property)
  - problems
    - Fractional knapsack
      - algorithm
      - run-time
    - Task scheduling
      - algorithm
      - run-time

- Divide & Conquer
  - philosophy (divide, recur, conquer)
  - problems
    - Merge Sort
      - algorithm
      - run-time
    - Integer Multiplication
      - algorithm
      - runtime
    - Recurrence equations and master theorem

- Dynamic programming
  - philosophy (subproblem optimality) (bottom-up)
    - subproblem overlapping (table)
    - define subproblems
    - show subproblem optimality
    - express solution to a larger problem through solutions to smaller problems
    - recurrence formula
    - implementation
  - problems
    - Knapsack problem
      - solution and algorithm
      - complexity
    - Matrix chain multiplication
      - solution and algorithm
      - complexity
Graphs

- Definitions
  - graph, vertex, edge, directed, weighted,
    vertex degree, adjacent,
    incident, path, simple path, cycle, simple cycle,

- Properties
  - subgraph, spanning subgraph, connected,

- Presentations
  - edge list, Adjacency list, adjacency matrix
  - performances

- DFS
  - algorithm (time bound)
  - properties
    - connected component of \( v \) by DFS(\( G, v \))
    - spanning tree by red edges
      (discovery edges, back edges = black)

  - applications
    - path finding
    - cycle finding
    - connectedness
    - connected components
    - spanning tree (forest)
    - bi-connected components
      - be able to find (any method)
      - separation vertices
      - separation edges
      - bi-connected components

- BFS
  - algorithm (time bound)
  - properties
    - connected component of \( v \) by BFS(\( G, v \))
    - spanning tree by discovery edges
      (cross edges)
    - layering the vertices of \( G \) \( l_0, l_1, l_2, \ldots \)

  - applications
    - connected components (connectedness)
    - spanning tree (forest)
    - cycle finding
    - path with min. number of edges
Directed graphs

- Definitions
  - in-degree, out-degree, directed path, reachability, directed cycle, DAGs, strong connectivity
- Representation
  - $v \rightarrow (\text{incoming edges})$
  - $v \rightarrow (\text{outgoing edges})$
- Directed DFS (complexity)
  - strong connectivity algorithm (complexity)
- Transitive closure
  - definition
  - algorithm (Floyd-Warshall)
    - running time
- DAGs and topological sorting
  - any topological sorting algorithm
    - running time

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-yes, neg. cycles-no)
- Shortest path in DAGs and linear time algorithm
  - algorithm (use topological sorting)
  - applicability (neg. edges-yes)
- All pair shortest path (Floyd-Warshall)
- Minimum Spanning Trees
  - definitions
    - Prim - Jarník's algorithm
      - algorithm
      - complexity
      - properties behind the correctness (partition property) (merge property)
    - Kruskal's Algorithm
      - algorithm (diff. from P-Y. approach)
      - data structure and implementation (find, union)
    - Borůvka's algorithm
  - NP-completeness
    - definitions
      - Dec. Problems
        - class \( P \)
        - class \( NP \)
        - class \( NP \)-hard
        - class \( NP \)-complete
      - idea of polynomial reduction
      - reduction from 3SAT to Vertex Cover
        (only construction)
      - some list of other \( NP \)-complete problems