

- **Coverage**
 - Lectures 18 - 24
 - Week 8, 11 - 13 Readings
 - Chapters 15, 16, 22 (covering Strongly Connected Components), 23, 24, 25 Discussions
- **Topics** (not exhaustive!)
 - Graph Problems: **SCC**
 - Pseudocode for SCC
 - Find solutions to SCC problems
 - Calculation of running time
 - Proof of SCC algorithm
 - Pseudocodes: **Dijkstra, Bellman-Ford, Floyd-Warshall, Fibonacci Numbers, LCS, Unbounded/Bounded Knapsack, Activity Selection, Scheduling, Huffman Coding, Prim and Kruskal**
 - Example: Write a pseudocode of ... any of the above
 - Example: Correct the following pseudocode...
 - Proving correctness using induction
 - Calculation of running time for all the above
 - Shortest Path Problems: single source using either Dijkstra or Bellman-Ford and all-pairs using Floyd-Warshall
 - Numerical solution with specific paths and cost values
 - NO Amortized Analysis (just skip it!)
 - Detection of negative cycles in both Bellman-Ford and Floyd-Warshall
 - Find numerical solutions to LCS, Unbounded and 0/1 Knapsack, Independent Set, Activity Selection, Scheduling and Huffman Coding
 - Both bottom up and top down implementations of **dynamic programming algorithms**
 - Minimum Spanning Tree: solving numerical problems with both Prim and Kruskal
 - Proofs of Lemma related to min cuts
 - If needed, Union-find data structure will be provided (you need to understand how it works only)