Assignment 1

Start Assignment

Due Oct 8 by 11:59pm **Points** 100 **Submitting** a file upload

This assignment covers greedy algorithms (Chapters 2-4 in the text). You have to typeset your solution in LaTex and use a compiled language for your implementation (e.g. no Python or JavaScript)

Problem 1: Cycle finding [50]

As part of this problem, you have to design and implement an algorithm to find a cycle (just one cycle) in an **undirected graph**. Specific tasks you have to accomplish are:

1. Design a correct algorithm and show it in pseudo-code [10]
2. Provide proof of the algorithm's correctness [10]
3. Find and prove the algorithm's running time [10]
4. Implement the algorithm in a compiled language and: [20]
    1. Write a graph generator (Hint: use an existing graph generation library if you can)
    2. Write test code to validate that the algorithm finds cycles
    3. Test the algorithm for increasing graph sizes.
    4. Plot the running time as a function of size to verify that the asymptotic complexity in step 3 matches experiments

Note: you cannot assume that the graph is connected.

Problem 2: Minimum Spanning Tree for "sparse" graphs

For this problem, we consider **undirected graphs** that have n nodes and at most n+8 edges. For these graphs, you have to design an efficient algorithm that finds the minimum spanning tree.

Specific tasks you have to accomplish are:

1. Design a correct algorithm and show it in pseudo-code [10]
2. Provide proof of the algorithm's correctness [10]
3. Find and prove the algorithm's running time [10]
4. Implement the algorithm in a compiled language and: [20]
    1. Write a graph generator (Hint: use an existing graph generation library if you can)
    2. Write test code to validate that the algorithm finds a spanning tree

   3. Test the algorithm for increasing graph sizes.
   4. Plot the running time as a function of size to verify that the asymptotic complexity in step 3 matches experiments

What to turn in

For this assignment, you have to turn in a .zip archive containing the following:

1. Latex source of your solution/report
   1. Use an algorithm package to typeset the algorithms in pseudo-code
   2. Use mathematical symbols throughout
   3. Embed graphs for the experimental results
   4. Organize the report in two sections, one for each problem
   5. Add a section for extra explanation on how to run your code and how testing and evaluation was performed
2. PDF compiled version of your report
3. Source code file for each solution