**Summer Coding Challenge 2023**

**Finale**

**Date and Time: 24th July 2023, 7:00PM-10PM**

**Submit the following by email to** [**mithun.kumar@snu.edu.in**](mailto:mithun.kumar@snu.edu.in)**:**

* + Solution zip file containing Algorithm, Runtime and Space complexity analysis, Screenshots of results.
  + Code file for problem A (We should be able to see the code and run it)
  + Code file for problem B (We should be able to see the code and run it)
  + Code file for problem C (We should be able to see the code and run it)

1. The state has created a portal that enables people to submit name of another person they want to complain about. Each person can have multiple complaints. Each complain would be a two tuple entry: <name of person, t name of person they are complaining about>. The task is to consistently label each person in the state as Good or Bad. The labeling may be wrong but not inconsistent. Your task to model this problem in the form of a graph problem. Propose an efficient algorithm to consistently label all the names as good or bad, or to decide that no such classification exists. Use the graph model you proposed in the previous part of the problem. Analyze the running time of the algorithm.

Solution:

Algorithm

Runtime and Space complexity analysis

Screenshots of results

<Attach code file>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Given two strings of lengths m and N, present an optimal data structure and algorithm that would allow you test whether m is a subset of N, if m <N. Analyze the runtime of the algorithm.

Solution:

Algorithm

Runtime and Space complexity analysis

Screenshots of results

<Attach code file>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. In a weighted directed graph G, find all-pairs shortest paths. Now identify k of these shortest paths, each of them starting and ending at unique start and end points (assume k start and k end points) and no two paths share any edges. If for a given graph, such a solution is not possible, the algorithm may report so.

Solution:

Algorithm

Runtime and Space complexity analysis

Screenshots of results

<Attach code file>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*