

Coding Assignment: 3

Theme: Dynamic Programming & Linear Programming

Rules:

- Use **Python 3.7** or higher in **Google Colab**. It comes with pre-installed packages like numpy, time, etc.
 - Submission should be a single **.zip** file (no other extensions like .rar, .gz) Keep this in mind.
 - Submission should contain
 1. Code files in **.ipynb** format. Use separate **.ipynb** files for separate questions. **DO NOT** use .py files. We will test your code in Google Colab.
 2. **Also, submit your solutions in the format ROLLNO_Qx.py for each question. Change Q suffix according to question number. This will be used for checking plagiarism.**
 3. README file describing how to run the code. TAs will run your code to see if they are really working.
 4. A write-up in **.pdf** format. It should contain answers to the questions with appropriate screenshots or test results from the output of your code. Treat this as a mini-report that TAs will evaluate after your code successfully runs.
 - **DO NOT** use built-in functions for cases where you have been asked to implement that exact same functionality. You can use built-in functions if you want to test whether your designed function produces the correct output.
-

Q1. Longest Common Subsequence

Given two strings str1 and str2, return the length of their longest common subsequence. If there is no common subsequence, return 0.

A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, "ace" is a subsequence of "abcde".

A common subsequence of two strings is a subsequence that is common to both strings.

Implement the LCS Problem using:

- a. Naive recursive approach (Top Down without memoization)
- b. Recursive approach with memoization (Top down with memoization)
- c. A Dynamic Programming Approach (Bottom Up)

Q2. Revised Simplex Algorithm

In your theory class and tutorial, you have studied the Tabular method to implement the simplex algorithm to solve linear programming problems. There is another algorithm called the Revised simplex algorithm: https://en.wikipedia.org/wiki/Revised_simplex_method.

The revised simplex algorithm is in essence equivalent to the simplex algorithm. The only caveat is that it makes the implementation part easy. Instead of maintaining a tableau, it employs a different matrix-based approach.

Read the code supplied with the Simplex Algorithm Tutorial and understand it carefully. Then modify it to perform the revised simplex algorithm as described in the Wikipedia link.

To test whether your implementation is correct, test it against the tabular simplex method using the same inputs.