Assessment

Theory of Algorithms, Summer 21/22

Due: last commit on or before May 8th, 2022

These are the instructions for the assessment of Theory of Algorithms in Summer 2021/2022. The assessment is worth 100% of the marks for the module. Please read the *Using git for assessments* [1] document on the Moodle page which applies here. As always, you must also follow the code of student conduct and the policy on plagiarism [2].

Instructions

The purpose of this assessment is to ensure that you have achieved the learning outcomes of the module while also providing you with sample work to show prospective employers. The overall assessment is split into the three interconnected components as detailed below. The percentages beside each heading give the weighting of each of the three components. You may assume that each bullet point has an equal weighting within its component. Note, however, that the examiners' overall impression of your submission may override the individual weightings where deemed appropriate.

GitHub Repository (20%)

Create a GitHub repository containing two Jupyter notebooks – these are described further down. The repository should contain the following.

- Clear and informative `README.md` explaining why the repository exists, what is in it, and how to run the notebooks.

- Comprehensive commit history, with each commit representing a reasonable unit of work.

Post Correspondence Problem Notebook (40%)

Include in your repository a Jupyter notebook called `post-correspondence.ipynb` that contains the following.

- Definition and explanation of the Post Correspondence Problem.

- Definition and explanation of the Bounded Post Correspondence Problem.

- Python function to solve the Bounded Post Correspondence Problem. The function should take two lists of strings and return `True` if they correspond, `False` otherwise.

- Explanation of what an undecidable problem is in computability theory, with reference to the Post Correspondence Problem.

Countdown Numbers Game Notebook (40%)

Include in your repository a Jupyter notebook called `countdown.ipynb` that contains the following.

- Overview and explanation of the Countdown Numbers Game [3].

- Discussion of the complexity of the Countdown Numbers Game [3].

- Python function, written in a functional programming style [4] to solve a Countdown Numbers Game. The function should take a list of six numbers and a target number and return a method to calculate the target from the numbers, if it exists.

- Explanation of the functional [4] aspects of your code.

More information about marking

In completing the assessment, you should consider the following four aspects of your work. It is important that your submission provides direct evidence of each.

Research

Evidence of research performed on topic; submission based on referenced literature, particularly academic literature; evidence of understanding the documentation for any software or libraries used.

Development

Environment can be set up as described; code works without tweaking and as described; code is efficient, clean, and clear; evidence of consideration of standards and conventions appropriate to code of this kind.

Consistency

Evidence of planning and project management; pragmatic attitude to work as evidenced by well-considered commit history; commits are of a reasonable size; consideration of how commit history will be perceived by others.

Documentation

Clear documentation of how to create an environment in which any code will run, how to prepare the code for running, how to run the code, and what to expect upon running the code. Concise descriptions of code in comments, README, and any documents or notebooks.

References

[1] I. McLoughlin, "Using git for assessments,"
https://github.com/ianmcloughlin/using-git-for-assessments/.

[2] GMIT, "Quality assurance framework,"
https://www.gmit.ie/general/quality-assurance-framework.

[3] "Countdown (game show) - wikipedia," 2022. [Online]. Available: https:
//en.wikipedia.org/wiki/Countdown_(game_show)#Numbers_round

[4] "Functional programming howto — python 3.10.2 documentation," 2022. [Online].
Available: https://docs.python.org/3/howto/functional.html