Project: Theory of Algorithms

ian.mcloughlin@atu.ie

Summer 22/23

These are the instructions for the project for Theory of Algorithms in Summer 22/23. The project covers 30% of the overall assessment of this module.

Submission

- The deadline for submission is March 31st, 2023.
- Your whole submission must be in a single GitHub repository.
- Use the form on the Moodle page to submit your repository.
- All you need to do is submit the repository URL.
- You should submit the URL as soon as possible.
- Commits in GitHub on or before the deadline will be considered.[1]

[1] Once you have submitted your URL, you do not need to do anything other than commit to your repository and push the changes to GitHub.

What to submit

- You must create a GitHub repository solely for this assessment.
- Your repository must contain a single Jupyter notebook called `polynomial_time.ipynb` on the topic of The Polynomial Time Complexity Class (P).
- You should imagine your classmates are the target audience for your work, explaining all concepts in terms you imagine they would understand.
- The notebook should contain text, code, plots, mathematical notation, graphics, and any other items useful in explaining the concepts.
- The notebook should be clean, clear, and visually appealing.
- Your GitHub repository must contain `README.md` and `.gitignore` files with appropriate contents typical of a GitHub repository.
- Your repository should not contain any necessary content such as temporary files.
- Your repository commit history should show regular and reasonably sized commits.

Marking Scheme

Your submission will be marked using the four categories below. To receive a good mark in a category, your submission needs to provide evidence of meeting each of the criteria listed under it[2].

[2] In line with ATU policy, the examiners' overall impression of the submission may affect individual marks in each category.

Research (25%)*:* evidence of research on topics; appropriate referencing; building on work of others; comparison to similar work.

Development (25%)*:* clear, concise, and correct code; appropriate tests; demonstrable knowledge of different approaches and algorithms; clean architecture.

Documentation (25%)*:* clear explanations of concepts in notebooks; concise comments in code and elsewhere; appropriate, standard README for a GitHub repository.

Consistency (25%)*:* tens of commits, each representing a reasonable amount of work; literature, documentation, and code evidencing work on the assessment; evidence of reviewing and refactoring.

Advice

- Students sometimes struggle with the freedom given in an open-style assessment.
- You must decide where and how to start, what is relevant content for your submission, how much is enough, and how to make the submission your own.
- This is by design — we assume you have a reasonable knowledge of programming and an ability to source your own information.
- Companies tell us they want graduates who can (within reason) take initiative, work independently, source information, and make design decisions without needing to ask for help.
- The point of this assessment is to demonstrate you can do that.
- You need a plan, you cannot just start coding straight away.

Policies

- You are bound by all ATU policies and any GMIT policies that have not yet been replaced by new ATU policies.
- Review the GMIT Quality Assurance Framework.[3]
- Pay particular attention to the Policy on Plagiarism and the Code of Student Conduct.
- If you have any doubts about what is permissible, email me to ask[4].

[3] GMIT. *Quality Assurance Framework*. `https://www.gmit.ie/general/quality-assurance-framework`.

[4] `ian.mcloughlin@atu.ie`