

Algorithmic essays

Michael Snowden

School of Computing and Communications
The Open University
Milton Keynes, United Kingdom
m.densnow@protonmail.com

Michel Wermelinger

School of Computing and Communications
The Open University
Milton Keynes, United Kingdom
michel.wermelinger@open.ac.uk

ABSTRACT

We define algorithmic essays as short reports with code and explanatory text that explore at least two approaches to solving a computational problem. Our aim is for students to practise skills like clear communication and evaluating alternative solutions, by writing essays to be shared with others. This poster presents our approach to an environment and resources that ease the writing and sharing of essays, with minimal setup requirements.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education; Employment issues**; • **Software and its engineering** → *Software libraries and repositories; Software testing and debugging.*

KEYWORDS

communication skills, clean code, collaborative learning

ACM Reference Format:

Michael Snowden and Michel Wermelinger. 2024. Algorithmic essays. In *The United Kingdom and Ireland Computing Education Research (UKICER 2024)*, September 5–6, 2024, Manchester, United Kingdom. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/3689535.3689547>

CS graduates may struggle with algorithm design and analysis, instead ‘hacking’ code and taking only one approach to solve the problem, without thorough testing [2]. They often also lack other skills highly sought by employers, like problem-solving, creativity, and communication [1]. In some HEIs, final year students acquire professional skills by contributing to open-source projects [3], but that takes significant effort from students and educators.

We wish to explore a simpler approach, for students in introductory programming and data structures and algorithms (DSA) courses: write essays with code, and provide feedback on peer essays. Explaining to others a problem and two alternative solutions, and evaluating their pros and cons (e.g. with respect to efficiency and simplicity), should help students consolidate their learning while practising communication, critical thinking and clean coding.

Writing an essay on a technical subject can be daunting. We therefore wished to provide not only guidance on how to write and structure algorithmic essays, but also essay examples and templates that follow the suggested structure and give students a starting

point to overcome writer’s block. We needed a platform that eases the writing and sharing of essays, in particular our examples and templates. We listed some requirements: free and easy to use; minimal software configuration and setup; support for commenting and publishing essays; support for code linting or other tools.

The most popular medium for weaving text and executable code is Jupyter Notebooks (JN) [4], making them the natural choice for algorithmic essays. We found 26 JN platforms, but their quality and suitability for our purposes varies. While cloud-based platforms satisfy most of our requirements, they often restrict access to the underlying notebook files, which complicates the use of linters and other tools. Moreover, their UIs tend to diverge from the classic JN interface and to include data science features, which increases the learning curve. Since no platform is ideal, after evaluating the options, we support Deepnote, Google Colab and local installations. In the latter case, students can share and comment on essays in their course’s online forum.

To accommodate different platforms, we organised and published our resources (<https://dsa-ou.github.io/algoesup>) so that students can start writing their essay by copying a template essay to their Deepnote or Google account in a few clicks. The template notebook links throughout to the writing and coding guidance for more details. Our aim is for students to read the guidance as they need it, during writing. Moreover, the templates include code that installs and imports a library we wrote to lint each code cell when it is run, and to show any messages directly below. This helps adherence to professional conventions during coding, not as an after thought.

While we target mainly DSA courses using Python, we believe more generally that a wide range of courses may benefit from interactive and multimedia essays or coursework reports, in the form of computational notebooks, to develop a range of skills. We have thus put our resources under an open source licence to ease adoption and adaptation.

ACKNOWLEDGMENTS

The work of the first author was funded by the Council of Professors and Heads of Computing with a Special Project grant.

REFERENCES

- [1] Quacquarelli Symonds Ltd. 2019. The Global Skills Gaps Report. <https://www.qs.com/portfolio-items/the-global-skills-gap-report-2019>.
- [2] Hisham Haddad. 2002. Post-Graduate Assessment of CS Students: Experience and Position Paper. *Journal of Computing Sciences in Colleges* 18, 2 (December 2002), 189–197. <https://dl.acm.org/doi/10.5555/771322.771350>.
- [3] Diomidis Spinellis. 2021. Why computing students should contribute to open source software projects. *Communications of the ACM* 64, 7 (July 2021), 36–38. <https://doi.org/10.1145/3437254>.
- [4] A. Rule, J. Landy, and M. Nichols. Research on the usage of Jupyter notebooks. <https://github.com/jupyter/notebook-research>.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

UKICER, September 5–6, 2024, Manchester, United Kingdom

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1177-0/24/09.

<https://doi.org/10.1145/3689535.3689547>