

Automatic Exercise Generation for Algorithmics Courses

Zoltan Sojtory

2442602s

Proposal

Motivation

With the start of the COVID 19 pandemic, schools and universities were forced to introduce at-home online examinations, which are still prevalent today. This presented new challenges in maintaining academic integrity, as it made monitoring student collusions very difficult. As a solution, each student can be handed unique exercise sheets, however these can be time consuming to design, so automatic exercise generation is the automation of this process. This method has worked successfully in many fields such as embedded systems and satisfiability problems but not much work has been done for algorithmics.

Aims

This project will develop software which automatically generates large numbers of uniform difficulty exercises based on some user inputs. More specifically, the exercises generated are a graph which tests Dijkstra's algorithm and a string which tests the Knuth-Morris-Pratt (KMP) algorithm. User inputs of certain characteristics are required to ensure uniform difficulty such as the number of vertices and edge-relaxations for Dijkstra and string length and longest border length for KMP. The effectiveness of the software will be validated through proofs for each generation algorithm as well as user testing with algorithmics lecturers.

Progress

- *Language for implementation: Solution to be developed with an object-oriented approach in Java, with JavaFX as the GUI framework if applicable*
- *Background research conducted on automated exercise generation in different fields, which was written up in background section of the dissertation*
- *Template based approach applied to Dijkstra and KMP to identify characteristics which impact difficulty*
- *Presented Motivation and Aims at a meeting for the Centre for Computing Science Education, who provided feedback*
- *Generation algorithms designed and implemented for Dijkstra and KMP*
- *LaTeX and PDF rendering implemented in Java for graphs and strings*
- *Simple console interface implemented*

Problems and risks

Problems

- *Tried to write up mathematical proofs for the generation algorithms, which proved to be very difficult due to the number of variables. Perhaps some kind of informal proof will be used instead.*
- *The free version of the API which renders LaTeX to PDF in Java creates a large watermark, so perhaps only LaTeX generation will be included in the final project, which can then be rendered separately.*
- *Generating exercises of equal difficulty is a very ambiguous task, so it was difficult to come up with a solution and the current solution may be refuted.*
- *Had some issues with the dissertation's bibliography not rendering correctly sometimes, which still happens.*

Risks

- *Algorithmics lecturers might be busy and thus difficult to get a hold of for user studies. **Mitigation:** other computing science lecturers should be sufficient, as the algorithms generated for are not very complex (Dijkstra and KMP).*
- *Writing the dissertation may take longer than expected, as I do not have too much experience with LaTeX. **Mitigation:** Read some LaTeX documentation perhaps over Christmas.*
- *Implementing solution generation (answer key) may take longer than the exercise generation algorithms did. **Mitigation:** Allow more time for completion in the first couple of weeks following Christmas.*

Plan

Semester 2

- *Week 1-3: finish developing solution generation with unit testing in Java*
 - **Deliverable:** Complete Java implementation which renders generated exercises as well as how to solve them and complete unit tests using JUnit
- *Week 4-5: finish proofs for generation algorithms*
 - **Deliverable:** Have formal proofs ready for the exercise generation algorithms for both Dijkstra and KMP, showing that they are accurate.
- *Week 6-7: user study*
 - **Deliverable:** Complete user study to be conducted and incorporated in dissertation.

- *Week 8: first draft*
 - ***Deliverable:*** *Finish as much of the dissertation as possible, to get the best feedback possible from supervisor.*
- *Week 9-11: final write up*
 - ***Deliverable:*** *Finish and polish dissertation and project to hand in.*