Prepared for submission to University of Cambridge

An implementation of a python sudoku solver package and a complete review of the software development process involved.

Adnan Siddiquei

University of Cambridge

E-mail: as3438@cam.ac.uk

Contents

1 Introduction

2 Selection of solution algorithm and prototyping

2.1 Solution algorithm

Why did I choose backtracking?

2.2 Prototyping

Put prototyping screenshots. Show plans of tests. Why did I choose those tests?

3 Development, Experimentation and Profiling

3.1 Linting and Formatting - `ruff`

Why did I use ruff as opposed to black and flake8? Faster, combines both into one. Can mimic both. PEP8.

3.2 Git and Gitlab pipeline

Why branches? Branch naming convention. Creating issues for tasks. Why? Project management. JIRA. New branch and merge per issue. Labels for issues because easy to and can folder the branches with those labels (show PyCharm example).

3.3 Test Driven Development

Explain why I wrote the tests first.

3.4 Profiling and Optimisation

Where did i test different packages for speed? Why did I choose numpy? Show how I profiled my code to identify where the bottleneck was. How did I work around this? Cython?

3.5 Coding Best Practises

Modularisation. typing. Exceptions. Error handling, try except. Never catch all exceptions.

4 Validation, Unit Tests and CI set up

Talk through why my unit tests are sufficient. How did I put this into the CI? With pre-commit.yaml

5 Documentation, Packaging and Usability

5.1 Documentation - `sphinx`

Why did I use sphinx over Doxygen? Sphinx is more popular, more support, more documentation, more features.

5.2 Packaging - Conda and Docker

What benefits does conda have over pip? Why did I choose conda? Why Docker?

6 Summary