Peer-review of assignment 4 for *INF3331-espenaf*

Reviewer 1, markuslh, markuslh@student.matnat.uio.no

October 9, 2018

1 Introduction

1.1 Goal

The main goal of this review is to provide **constructive feedback and advice** on how to improve the solution.

1.2 Guidelines

You, the peer-review team, can decide how you organise the peer-review work between you. For instance, you can work as a team, writing all report together, or you can split the work and write each one report.

For each (coding) exercise, one should review the following points:

- Is the code **working as expected**? For non-internal functions (in particular for scripts that are run from the command-line), does the program handle invalid inputs sensibly?
- Is the code **well documented**? Are there docstrings and are the useful?
- Is the code written in **Pythonic way** [1]? Is the code easy to read? Are the variable/class/function names sensible? Do you find overuse of classes or not sufficient use of functions or classes? Are there parts of the program that are hard to understand?
- Can you find **unnecessarily complicated parts** of the program? If so, suggest an improved implementation.
- List the programming parts that are not answered.

Use (shortened) code snippets where appropriate to show how to improve the solution.

1.3 Points

The review is completed by pushing the review Latex source file (.tex files) and the compiled PDF files to the repository of the students that are reviewed. The name of the files should be *feedback.tex* and *feedback.pdf* in the students assignment4 directory.

You will get up to 10 points for delivering the peer-reviews. Each of you should contribute to the review roughly equivalently - your team will get the same number of points[2].

2 Review

This review was made on Ubuntu 18.04.1 LTS with Python 3.6.6.

Assignment 4.1

Is the code working as expected?

The code works as described. However it wouldnt give me the picture in .jpg-format as described by `--help` By adding some default values the program would also be easier to run

Is the code well documented?

Good docstrings.

[1] https://www.python.org/dev/peps/pep-0020/

[2] In case a team-member does not contribute, please email simon@simula.no

Is the code written in Pythonic way?

The code seems to be well written with good variables and not too dense.

Can you find unnecessarily complicated parts of the program?

Nothing out of the ordinary, seems fine.

Assignment 4.2

Is the code working as expected?

As for Assignment 4.1

Is the code well documented?

As for Assignment 4.1.

Is the code written in Pythonic way?

As for Assignment 4.1.

Can you find unnecessarily complicated parts of the program?

No.

Is numpy being used effectively (i.e. vectorization where possible)?

Vectorization seems good.

Does the report match the output when run?:

- Report on runtime (`report2.txt`) with parameters used and comparison to mandelbrot_1 and/or mandelbrot_2 is actually a bit faster on my computer but thats probably because of different hardware. The two seem to be close to equally fast and not much time has been saved with numpy here

Assignment 4.3

Is the code working as expected?

As for Assignment 4.1. Numba increases the overall speed significantly

Is the code well documented?

As for Assignment 4.1.

Is the code written in Pythonic way?

As for Assignment 4.1.

Can you find unnecessarily complicated parts of the program?

No.

Does the report match the output when run?:

- Report on runtime (`report3.txt`) with parameters used are much faster compared to mandelbrot_1 and mandelbrot_2

Assignment 4.4

Not delivered.

Assignment 4.5

Is the code working as expected?

The UI works as described. Suggestions for improvements:

- As for assignment 4.1 maybe adding some default values so the program could run without having to input values would be nice
- Maybe giving some help as to how to run the program without having to type `--help` could be useful, but not needed

Is the code well documented?

Good docstring and well documented.

Is the code written in Pythonic way?

Seems pythonic.

Can you find unnecessarily complicated parts of the program?

No but maybe import packages could be an alternative as some of the code is re-used throughout $\texttt{mandelbrot}_1 - 3.py$

Assignment 4.6

Is the code working as expected?

`File submitted but no code found.`

Does the setup.py script install the module/package containing the "compute_mandelbrot" function?

`No.`

Are the tests well designed and do they have a meaningful name?

`NA.`

Assignment 4.7

`Delivered---the comments for assignment 4.1 apply.`

Assignment 4.8

`repeat_good() Clever usage of .title(). Concise. repeat_bad() Unnecessary amount oc computing spent on the modulus-calculation, nice and ugly.`

General feedback

- `Overall nice solutions and readable code which makes it easier to follow what happens inside the program itself`
- `More intuitive help when starting program`
- `Default values to run program if run with no user-input`