SNACS 2021 Code Review Best Practices

General PSA’s

- If code implemented in NetworkX try SNAP
- Use LIACS computers for very large graphs

Code practice

- Use clear, modular functions to reuse code (52+27)
- Clean code
- Try to keep functions short and clear (e.g. by dividing them up)
- write informative docstrings about inputs, outputs and what the function does
- Write clear, and self-explanatory variable names.
- Write meaningful error messages (e.g. use catch/except + raise exception)
- Write comments so that your teammates can easily understand the code
- Generally write readable code that can be expanded upon by other users, without them having to study your code for a very long time. (52+27)
- When not using Python, still mind your indentation. (52+27)
- If possible sort before a loop
- comment your code
- Keep your code DRY (Don’t Repeat Yourself)
- Avoid Deep Nesting.
- readability is more important than compact code (e.g. avoid long one-liners)
- Think of random seeds when you’re conducting experiments

Python tips

- Use multiprocessing, e.g. Pool to speed up code if your algorithm allows for it (1+2)
- Look at dataclasses or NamedTuple to prevent ‘magic’ indices throughout your code (1+2)
- Use Python typehints (https://docs.python.org/3/library/typing.html)

Plotting

- use seaborn for nice plotting (e.g. set_style(“whitegrid”))
- Don’t forget to put labels for axis!!!
- use bold face + larger lettering for axes for clarity
- xticks for changing tick labels/ frequency
- save plots as pdf for high res in latex documents
- make your colormaps color-blind friendly

Safety from frustration

- Use GitHub to collaborate on code (or something similar).
- Use Github to be able to revert to previous versions of the code! (3+41)
- use Google colab to easily program together.

- Add comments in your code
- Use the VSCode Live Share feature to collaborate on any code (1+2)
- come up with a nice file structure to keep everything organised
- If you’re running an algorithm that’s taking 5+ hours or so, instead of completely discarding it, maybe it’s possible to cap the time to 1 hour and if it takes longer you stop running the algorithm and just use what you get from that hour.
- Don’t use jupyter notebook for heavy tasks (we encountered some problems)

VScode tips

- Use a formatter (e.g. ESlint)
- https://code.visualstudio.com/docs/getstarted/keybindings
- nice extensions:
    - remote ssh
    - bracket pair colorizer
    - indent-rainbow
    - live-share
    - Github Copilot