

description

My structure

I Have delivered the two versions of this code that was produced during the lab.

In the folder `Assignment2_part1`, is the code for the Australia problem, and in `Assignment2_part2` is the code for the South America problem.

What work did I do?

I made a few improvements on the template, before starting the work on the Assignment.

That work is detailed in the first section below, while the section after that details the work made to get the template working, and the last section details the changes made to the template to fit the South America problem.

Template improvements

To improve upon the template I added an Enum to hold the States, and the colors.

This allowed me to get autocompletion when filling in specific countries, and most importantly it allowed me to do strict (and might I add, proper) type hinting for all methods, which was also added.

Getting the template to work

I implemented the methods based on the theory from the classes, and applying that to the methods, based on their names, and trying to piece together where they go, again based on their names.

Following that I implemented the pseudocode provided in the assignment. I did struggle quite a bit initially, trying to figure out what data types, go into the assignment variable.

In the end I figured out that the assignment should be a dictionary, where the keys are States, and the values, are the color assigned to that state.

After that realization though, the implementation went rather smoothly. I only needed to add the `__lt__` and `__hash__`

methods into the States Enum, so they could be sorted, and used as keys in a dictionary respectively.

After this the colors were correctly assigned, when the code is run.

Modifying for South America

To modify the code to work with south america, I added another color (Yellow) into the Color enum.

I improved a bit on the assignment of variables, domains and constraints to the CSP object.

I realized that they get assigned all the values from the enums, or get the same thing applied to all elements of a list.

This allowed me to automate it, which was very nice. I then spent some time on getting the neighbours written in correctly.

This was however made quite a lot easier by the Enums aiding with autocomplete.

I chose to write in the countries from the image ordered from the top down in a scanning fashion, when I wrote the neighbours in, I did so in a clockwise order starting from the top.

Following these changes the solution was correctly coloring the map, but to make it easier to check I added another print statement in the output loop, which shows the neighbours of each state and their colors. Feel free to use it if you check my solution, the countries are printed in the order described above, and the neighbours are printed roughly in that order as well :)

GitHub Repo

The code can also be found in my GitHub repo <https://github.com/thor2304/AI-Course>.

The delivered materials, can be found in the folder: **Lab6**.