1. Describe any implementation choices you made that you felt were important. Clearly explain any aspects of your program that aren't working. Mention anything else that we should know when evaluating your work.

While I believe the overall implementation of my fc and mcv algos are on the right track, they don't seem to be accelerating the overall solution time. I believe this is most likely due to the implementation of storing neighbors I chose and issues with pruning said neighbors.

2. What is the size of the state space for this problem?

Given a map it can be said the maximum number of colors needed to color said map is the maximum number of edges any vertices has plus 1. Given that the coloring configuration is non-constrained this means each configuration can be applied n ways n times. With this in mind an upper bound on the state space, assuming the state space is all possible coloring configurations can be: (max_number_edges_of_any_single_vertice +1)^n

3. What's the average speed-up you get for fc over dfs? For mcv over fc?

Due to my implementation fc preforms similarly to DFS but MCV takes longer consistently

4. (830 only) Discuss the effectiveness of randomized restarting.

5. What suggestions do you have for improving this assignment in the future?

None