

Group member: Amani Reed

## **Report for Assignment 1**

I started this assignment by using a scanner object to read in the input file with the colors, the states, and the list of adjacent states. In order to gather the information, I used regular expressions to scout out the colors (the first section), the list of states (the second section) and the list of adjacent states (the third section). For example, for the colors, I knew that all the colors were given in the input file with the first letter uppercased and the rest of the letters lowercased, so the regular expression I used was “[A-Z]+[a-z]+” to look for one or more occurrences of a string that begins with an uppercase letter followed by lowercased letters. I used a similar method to read in the list of states and the list of adjacent states as well. After I did this, I stored a list of the colors (first section of input file), the list of states (second section of input file), and the list of adjacent states (third section of input file) in an array list, called “listofcolors”, “listofstates”, and “listofadjacentstates” respectively. I then broke up the third section of the input file (thelistofadjacentstates arraylist) into two columns so I could reorganize the information provided in a way to see all the adjacent states for each state. To do this, I used a hashmap and put the list of states as the key and initialized an arraylist as the value in order to have a list of adjacent states for each key. Once I did this, I attempted my backtracking search by selecting a non-colored node from the graph. For each possible coloring of the node, I checked if the combination (node, color) satisfies the constraints (if two states are adjacent, then they should have different colors). I made sure that if two states were adjacent to each other, then they should be assigned different colors, and once these conditions were met, I assigned different colors to the adjacent states. After this, I attempted my local search by having an initial node and an

adjacent node, and if the value of the initial node was higher than the value of the adjacent node, then I would use the current node.

For this assignment, I also had a node class that took in the state and its color and had getters for these attributes.

I worked alone on this assignment, I did all parts by myself.