John Guttadauro
28160001

My program files are in the Graph sub directory in my edlab folder. To run either of the large graphs, give the flag -Xss12m to increase java stack size. The output from both the colorable and not colorable graphs were more than 100 entries long, and are in the Graph sub directory under the names largegraph1output.txt and largegraph2output.txt.

For this assignment, I intake the given file using buffered reader, and build an adjacency list as I intake each line. For each numbered vertex I intake, I insert it at the number minus one in the array by creating a vertex object that holds a list of the neighbors of the point. Adding to the list of neighbors is a constant time operation, as a tail pointer is maintained and replaced as new entries are added. Building this list takes E time, as it is built from reading all the edges. Nodes that have no edges are not created at this stage, and are dealt with later on.

The program then loops through the list of vertices, at each entry checking if the vertex has been colored. If the vertex is not colored, the program assigns it a color, and then executes a recursive breadth first traversal of the graph, while coloring all nodes reachable from the starting point. If the vertex is colored, it is considered visited and ignored. This process takes O(|E| + |V|) time, as it checks the edges of each vertex during a breadth first coloring call, and the overarching loop goes through each vertex. If the coloring method attempts to assign a node a color that is opposite of what it is already assigned, this is considered not two colorable. The recursive call marks all nodes that are reachable from this conflict as part of the substructure that causes it on the way back up the stack.

The output of this program depends on whether or not the graph was two colorable. If it was, the output method visits each vertex and outputs its number, neighbors, and color to the file. This takes O(|E| + |V|) time, as it prints each vertex and all its edges. If the graph was not two colorable, the program still walks through the entire list of vertices, but only prints the number, color, and neighbors of any members of the non-colorable substructure. This would run in O(|V| + Not Colorable(|E|)) time.

Reported runtime from edlab machine
largegraph1:
- real: 0m1.861s
- user: 0m3.648s
- sys: 0m0.232s

largegraph2
- real: 0m2.218s
- user: 0m3.764s
- sys: 0m0.264s

smallgraph:
- real: 0m0.115s
- user: 0m0.096s
- sys: 0m0.012s