**AI-Project-1: CSP (Graph Coloring)**

**Student Name:** Javid Guliyev

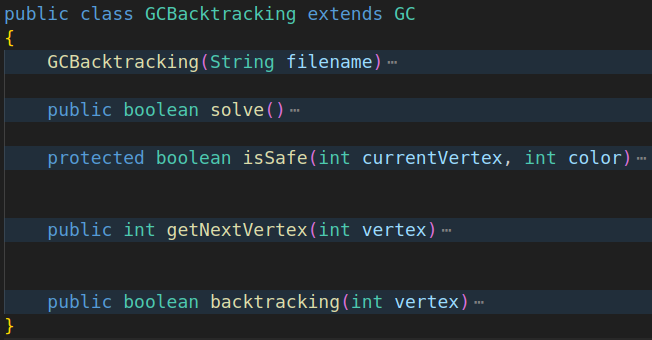
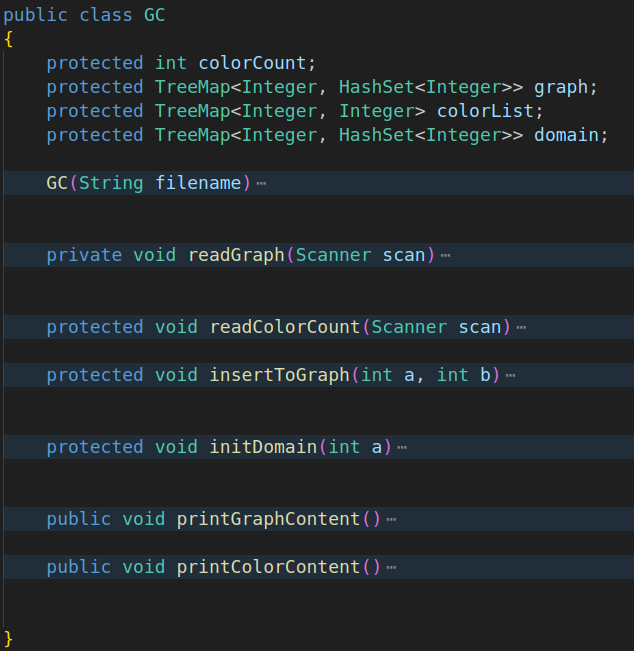
**Student ID:** g34987427

**Introduction:**

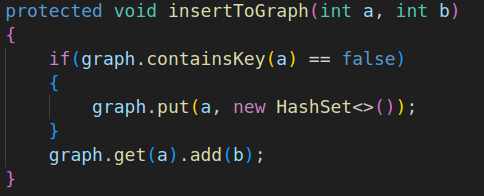
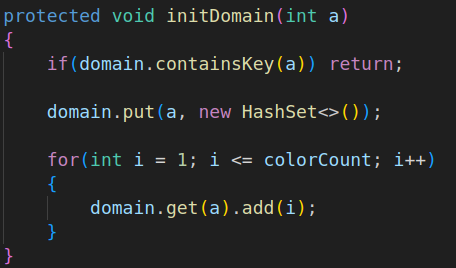
This report outlines the structure and key concepts behind the implementation of a solution for the Graph Coloring problem in the context of Constraint Satisfaction Problems. As with previous projects, the chosen programming language is Java. The solution will utilize a **backtracking** approach with **ordering heuristics**(MRV, LCV, Tie-breaking) and the **AC3 algorithm for constraint propagation**.

**General Structure**:

As it can be observed the there couple of classes that contains different approaches to tackle this problem, however they all are the child classes two base classes structure of which are provided below:

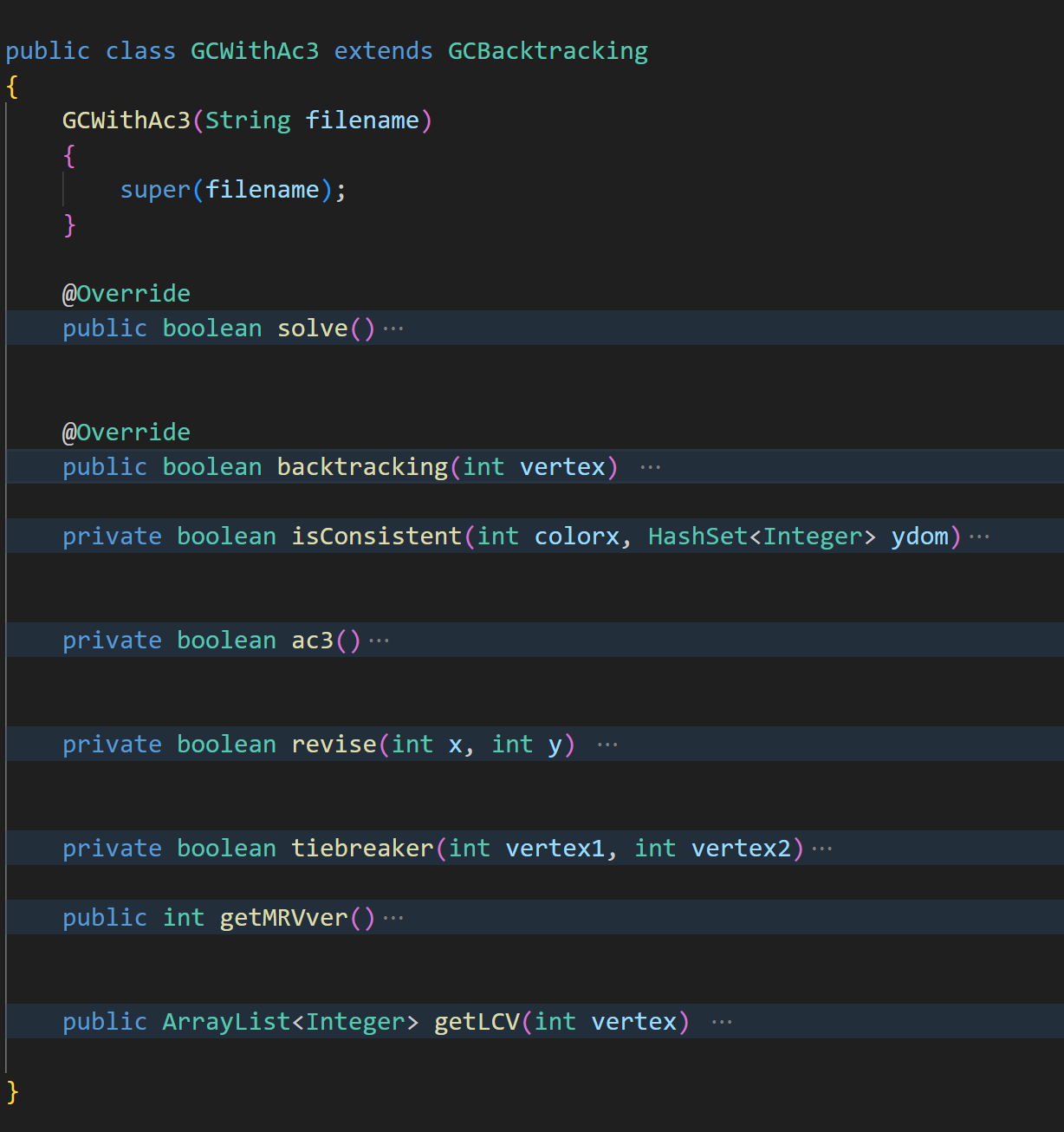


The first class (on the left) is used only to read contents of the files to graph, determine color count and initialize domains. The TreeMap is preferred over the HashMap as keys are stored in sorted order.



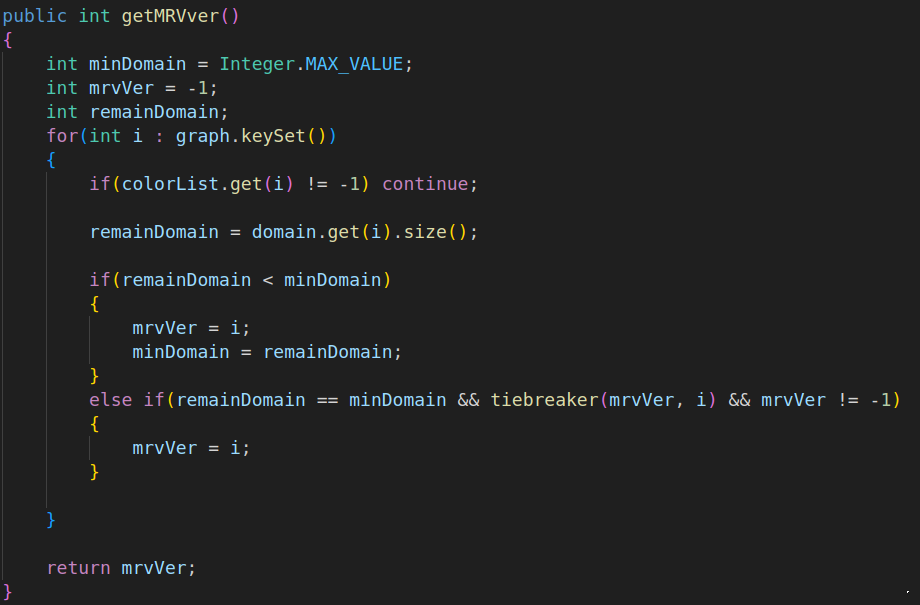
The second class uses simple backtracking algorithm to solve the graph coloring. The other classes will extend and override backtracking by adding ordering, filtering and constraint propagation techniques.

The main class the report will focus on is the AC3 with ordering approach which is contained in its own separate class:

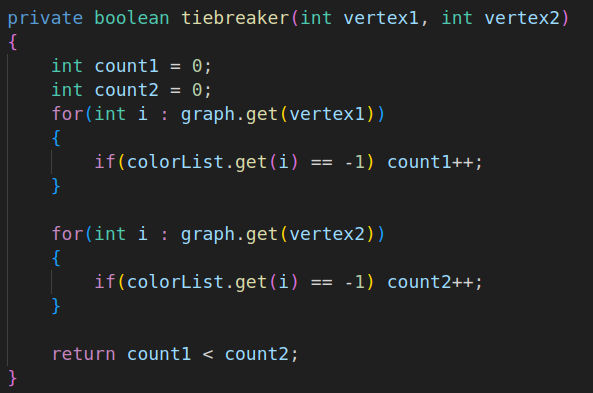


**Heuristics(Ordering):**

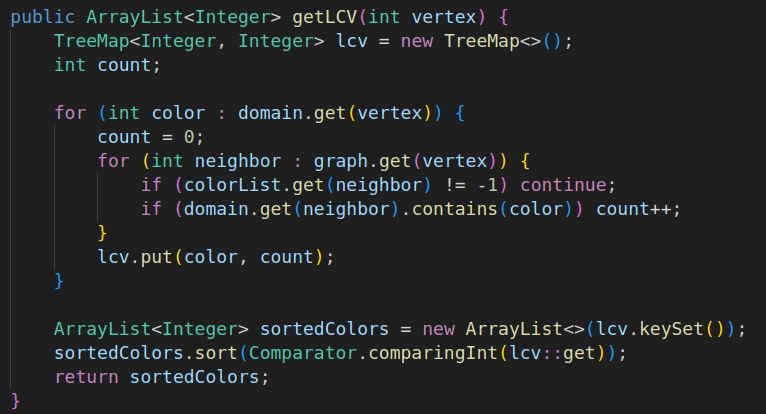
**MRV:** the first algorithm used to enhance the backtracking is MRV which will loop through the all the domains of the vertexes that are not yet assigned and find the vertex that has the least amount colors in their domain. If the all vertexes are assigned the method will return -1



**Tie-breaker:** In the early stages of backtracking, it is common for multiple vertixes to have the same number of values in their domain. To maximize efficiency, it is recommended to select the vertex with the highest number of unassigned neighbors(in other words vertex that is involved with more constraints).



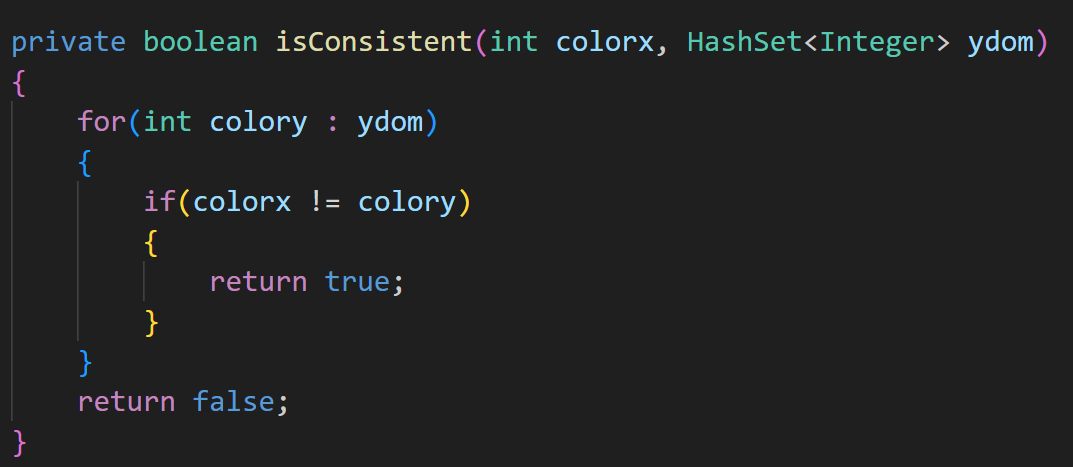
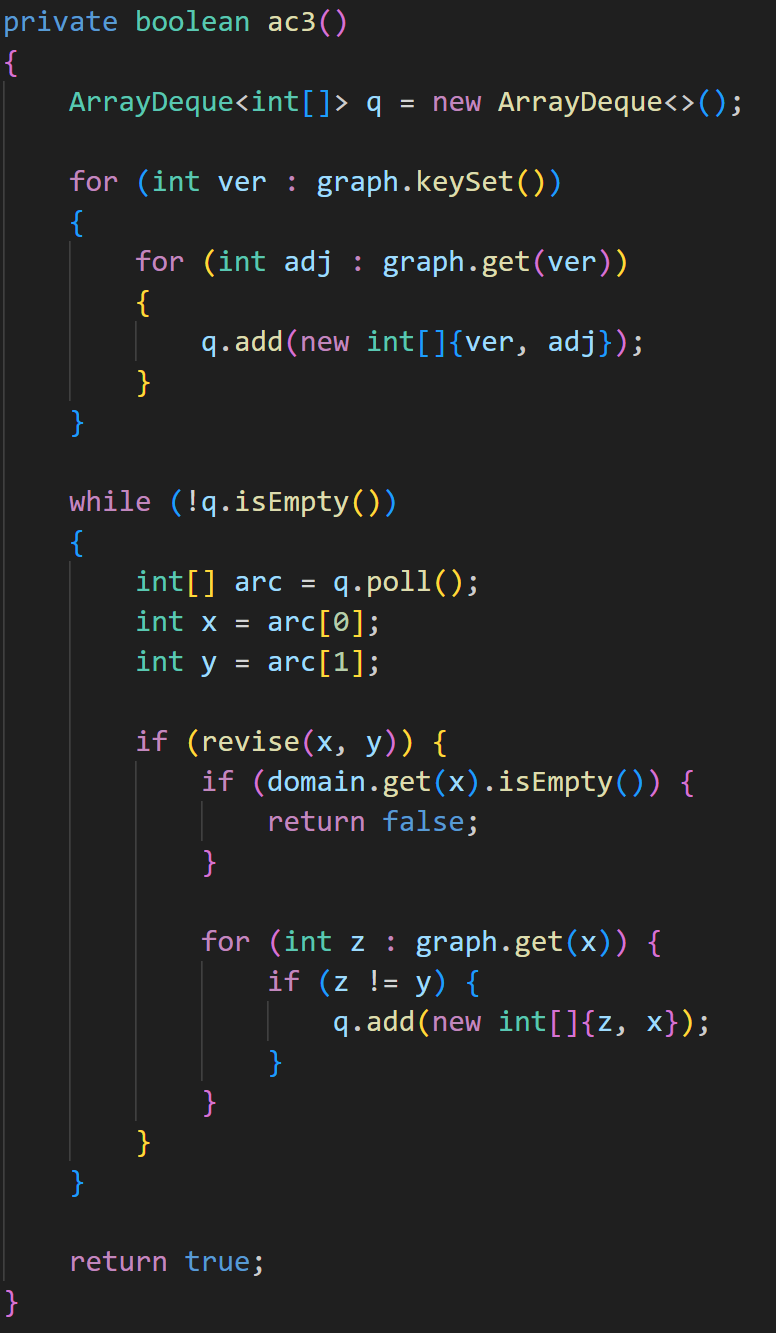
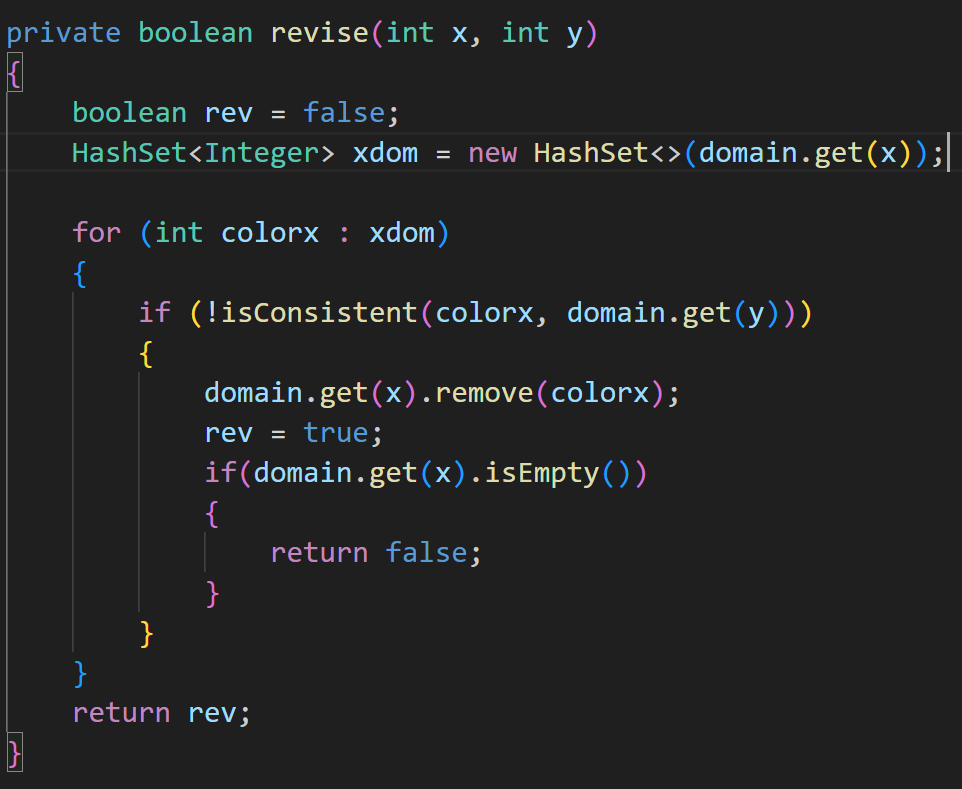
**LCV:** Now as the vertex is selected it is important to decide which color to choose from its domain. LCV offers to select the color that will effect domains of neighboring vertexes less. LCV will return the list of colors sorted from least constraining to most constraining.



Those 3 methods alone can significantly boost the performance simple backtracking by providing effective choice of vertexes(Tie-breaking, MRV) and selecting the colors that will reduce the amount of backtracking in general.

**Constraint Propagation (AC3):**

**AC3:** Constraint propagation using AC3 algorithm to reduce the domains of vertexes globally resulting in less backtracking. The AC3 ensures that the graph is arc-consistent by iterating over all arcs (pairs of connected nodes) in the graph, stored in a queue. For each arc, it calls the revise method, which reduces the domain (possible values) of the first node (x) based on the values of the second node (y). If a domain becomes empty, the method returns false, indicating inconsistency. Otherwise, it re-adds the modified arcs to the queue and continues the process until no further revisions are needed, ultimately returning true if the graph is arc-consistent.



**Revise and IsConsistent**: The revise method aims to ensure arc consistency by iterating over all possible values (colorx) for a node x and checking if they are consistent with the values of an adjacent node y. If any value of x is found to be inconsistent via IsConsistent method with all values of y, that value is removed from the domain of x, and a flag (rev) is set to true. If the domain of x becomes empty after removing inconsistent values, the method returns false, indicating a failure in maintaining consistency. Otherwise, it returns the value of the flag, indicating whether any changes were made to the domain of x.

**Putting all together (AC3 + Backtracking + Ordering):**

The backtracking method is used to solve a constraint satisfaction problem by recursively assigning colors to vertices in a graph. The method tries each color in the least constraining value (LCV) order. For each color, it makes a deep copy of the domain to preserve the original state, assigns the color to the vertex, and checks arc consistency using the AC-3 algorithm and propagates constraints to other vertexes. If the AC-3 algorithm maintains consistency, it proceeds to the next vertex determined by the minimum remaining values (MRV) heuristic and recursively calls the backtracking method. If a solution is found, it returns true; otherwise, it backtracks by restoring the original domain and removing the color assignment, ultimately returning false if no valid solution is found.

