Ethan Chou - 301422045 - September 24, 2025

The encoding of my program involves the creation of an additional "Edge" class to help manage the edges (and vertices) passed into the program via the input.txt file. Java tidbits not related to course content and Z3 were adapted from w3schools.com.

Initially, my intention was to create a 2-D array of Expressions to represent the boolean variables of $p_{v,c}$ as per the hint given; similar to the in-class pigeonhole example. However, I found that by setting colour constraints on all colour(v) values to be within 1 and M that only one-dimensional IntExpr array was required so I went with that. By altering my approach it almost certainly led to a more efficient and not as complex program design. It also indirectly sets the constraint that each vertex can only have one colour since it is a 1-D array and we can't assign multiple values to the same spot in the array.

To ensure that no connected vertices had the same colour, I just had to add another constraint for each of the edges ensuring that the colours of the vertices on either end don't match.

The formula I used for constraining that every vertex is coloured and that every vertex is at most coloured is $p \wedge q$, where $p \leftrightarrow (colour[v] \geq 1)$ and $q \leftrightarrow (colour[v] \leq M)$.

The formula used for constraining that no connected vertices were the same colour is $\neg (C_{u,c} \wedge C_{v,c})$, where for all Edge(u, v), $C_{u,c}$ is the colour of vertex u and $C_{v,c}$ is the colour of vertex v.