

## BONUS Assignment – Vertex Coloring using SAT

Due Friday, 17 May, 11:30pm

**Part 1** Implement in Python a program that takes as input a graph  $G$  and a positive number  $k$ , and, based on the propositional logic description of the Vertex Coloring problem presented in the course slides, uses a SAT solver to check whether the vertices of  $G$  can be colored using at most  $k$  colors.

**Part 2** Implement in Python a program that takes as input a graph  $G$ , and returns its chromatic number  $\chi(G)$ , using your Python program from Part 1.

**Input** Consider the following format to describe undirected graphs. In this description, the lines that start with letter `c` are comments. The line that starts with `p` describes the number of vertices and the number of edges. The lines that start with `e` describe the edges.

```
c FILE: example.col
c DESCRIPTION: an undirected graph
p edge 9 13
e 1 2
e 1 4
e 1 7
e 1 9
e 2 3
e 2 6
e 2 8
e 3 5
e 3 7
e 4 5
e 4 6
e 5 8
e 5 9
```

**Submit** You are expected to submit a zip file containing 1) a 1-slide presentation (in PDF) at SUCourse+ illustrating a workflow of your Python programs, 2) your Python programs, and 3) two benchmark instances and their outputs for each part. Incomplete submissions will not be graded.

**Demos** Provided that your submissions are complete and your programs run without any bugs, we will grade your solutions based on your demo (including your presentation and your explanations to our questions during the demo). A submission without a demo will be graded as 0. The demos are planned for the week of May 20, and will be scheduled later on.

**No collaboration is allowed, individuals should submit their own solutions.**