CS270: LAB #19

Graph Coloring

You may work in teams of ideally one or two people (three is acceptable in the event of an unscheduled absence). Unless stated otherwise, the lab is due to be submitted into Gradescope at the end of the week (Friday 11:59pm)
In order to receive credit, follow these instructions:

[a] Every team member should be discussing simultaneously the same problem – do NOT try to divvy up the labor and assign different problems to different students since the material is cumulative.

[b] Directly edit this lab PDF using Sedja/PDFescape with your answers (extra pages can be added in the rare event you need more than the allotted space)

[c] Each lab, rotate which member has the responsibility of being the Scribe. This is the person that is typing the answers and uploading the final PDF – note that only a single copy of the filled in PDF is turned into Gradescope. Only one lab needs to be submitted for the entire team, and all members receive the same score. Make sure to use a font that your PDF editor is compatible with (otherwise you might find your answers appear as weird shapes/sizes or simply disappear entirely!)

[d] The Gradescope submission must have each answer properly tagged with the appropriate question. Moreover, every member of the team must be listed as a submitter. Although it is the Scribe which executes these actions, it is still the responsibility of the entire team to make certain this is done properly (thus it is highly recommended that the Scribe share their screen so the entire team can witness it). Answers which are improperly tagged cannot be seen by the grader and thus cannot be scored.

[e] **FOR REMOTE ONLY**: Each lab, rotate which member has the responsibility of being the Recorder. This is the person who hits the Zoom Record button (once the technical permission is granted by the TA/RCF/Professor) and ensures that everyone has their camera/microphone on. They are also the member that is responsible to make sure the DrexelStream video is marked as viewable and entered into the https://tinyurl.com/VidLinkForm webform before 11:59pm (they should also email the rest of their team as confirmation.) Note that the video file doesn't get created/processed until after the Recorder has quit Zoom.

[f] Each lab, rotate which member has the responsibility of being the Manager. This is the person that ensures that everyone is participating equally and honestly, keeps the group on task, ensures that all team members understand a solution before going on to the next question, and presses the "hand up" button in Zoom to summon a TA or the professor (but they only do so after surveying the group to make sure everyone has the same question).

Team Name (CS pioneer): Marvin Minsky

Scribe name: Terie Ha

Recorder name: Evelyn Thai

Manager name: Lixiao Yang

Other team member (if any): Jerry Li

Question 1 : 6 points

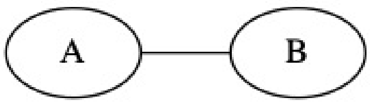
2 3 Color Problem

Any Boolean Formula can be satisfied by coloring in a graph. This is called the 3-Color Problem.

In this section, you will see how a Boolean Formula is related to a Graph Coloring.

2.1 Rules of the 3-Color Problem

The below imagine contains a graph. The circles are called **nodes** and the lines are called **edges**. Each **node** may have a **label** written inside it.

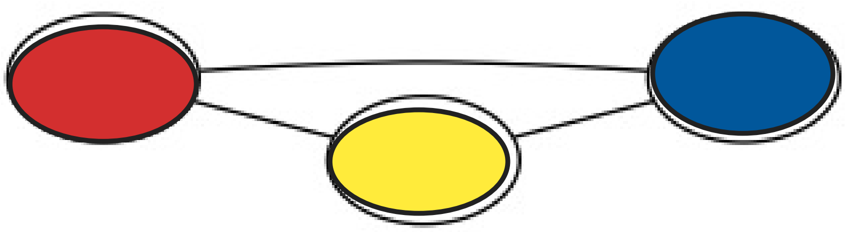


In the 3-Color Problem, we want to color in the **nodes**. We have exactly three colors. For example, we could use red, yellow, and blue. If two nodes are connected by an edge, they cannot be the same color. The goal is to color all the nodes in.

- Select 3 Colors. Use the Shapes[border] command available in Sejda/Word/Paint etc. Use actual colors (not text!)
- Try to color in all the nodes.
- If two nodes are connected by an edge, they **cannot** be the same color.

We will show in this exercise that if you can color in the nodes, it will tell you the true/false values of the variables in a Boolean Expression. If you cannot, it means the Boolean Expressions is contradictory.

Color in the following graph using the rules of the 3-Color Problem. The color you pick for node T means true and the color you pick for node F means False. Be consistent about colors in future questions.



(a) (2 points) Color in above graph correctly.

(b) (2 points) What Color did you pick for **True** (the T node)?

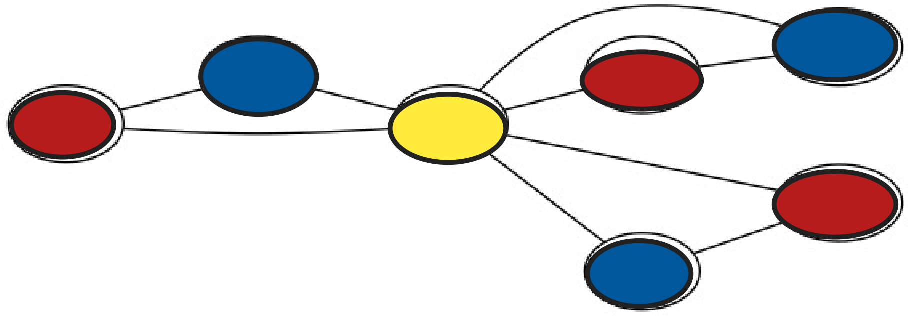
Red

(c) (2 points) What Color did you pick for **False** (the F node)?

Blue

Question 2 : 14 points

Color in the following graph using the rules of the 3-Color Problem.



Question 3 : 12 points

Select the Boolean value assigned to each node label by your coloring.

These answers should be clear from your coloring. For example, the node "not x" should match the color of either the T node or the F node. If it's color matches the T node, then we know $\neg x = True$.

(a) (2 points) x

- ● True
- ○ False

(b) (2 points) ¬x

- ○ True
- ● False

(c) (2 points) y

- ○ True
- ● False

(d) (2 points) ¬y

- ● True
- ○ False

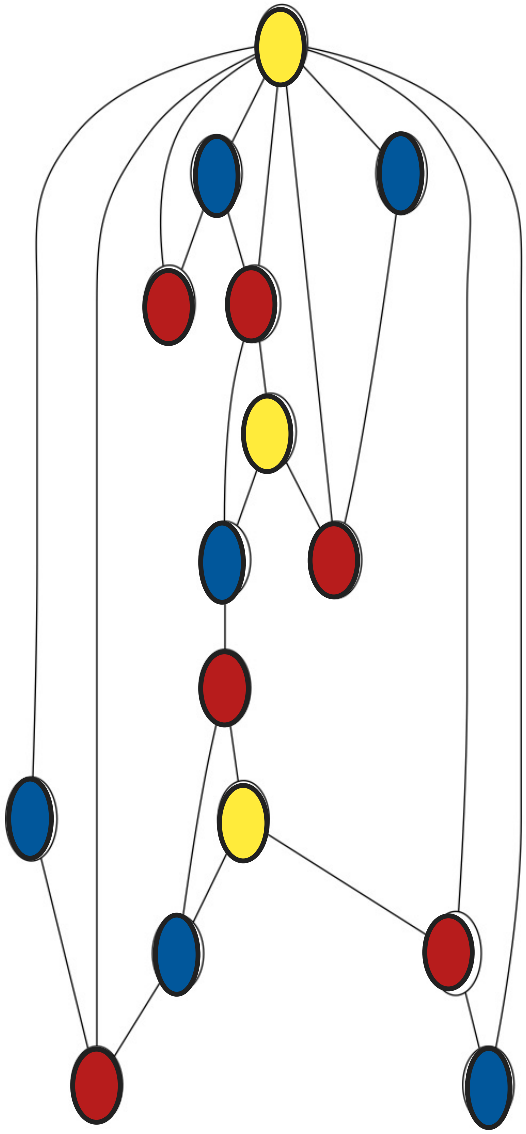
(e) (4 points) In the above graph, what color did you apply to the N node? A variable node is a node labeled with a variable (x and y in this example). Is it possible for any variable node to have the same color as the N node?

We applied yellow to the N node.

It is not possible for any variable node to have the same color
as the N node because the variable nodes should have the same colors
as True or False.

Question 4 : 8 points

Color in the following graph using the rules of the 3-Color Problem.



Question 5 : 6 points

Select the Boolean value assigned to each node label by your coloring. Use the coloring above to answer these questions. The answers should be clear from the colors you assigned the nodes.

(a) (2 points) x

- ● True
- ○ False

(b) (2 points) $\neg y$

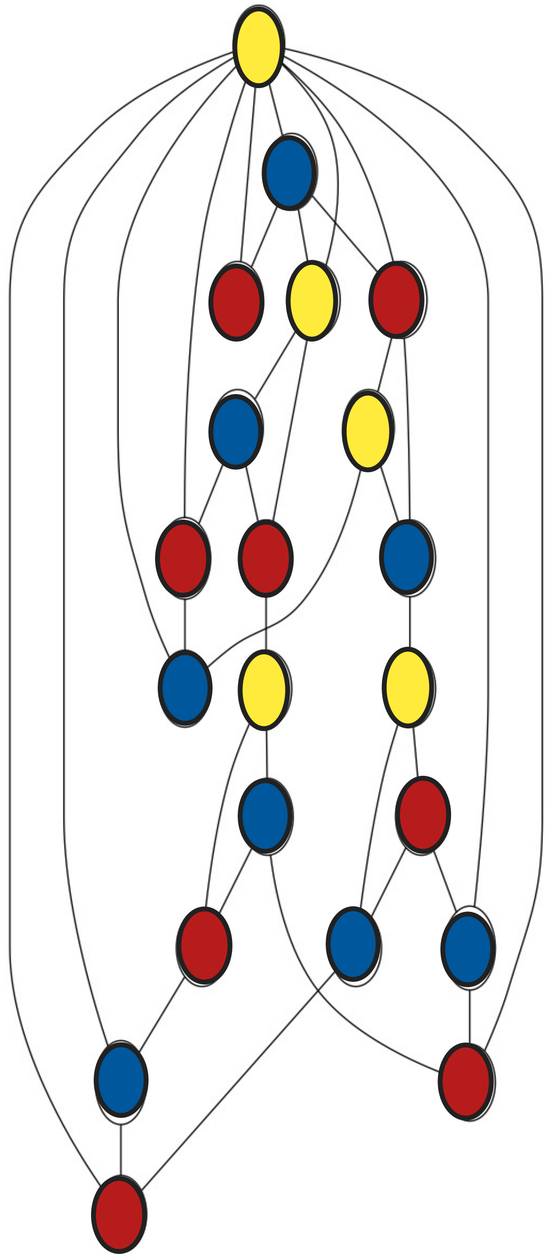
- ● True
- ○ False

(c) (2 points) z

- ● True
- ○ False

Question 6 : 8 points

Color in the following graph using the rules of the 3-Color Problem.



Question 7A : 6 points

Give the Boolean values that make the expression below true.

If you found a coloring of the graph that worked, that coloring is the answer to this question. Just match the colors with the variable values to answer this question.

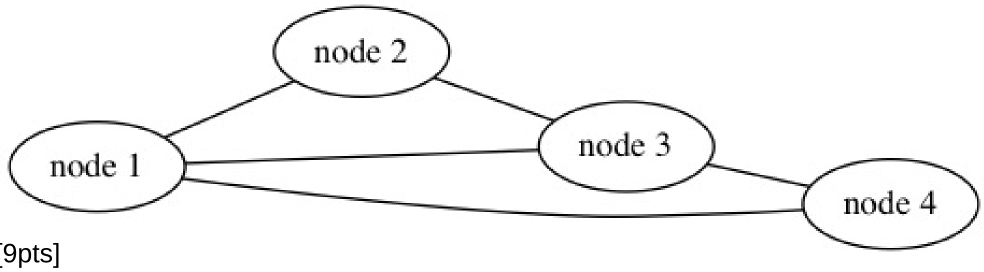
$$(\neg y \lor x \lor z) \land (\neg x \lor y \lor \neg z)$$

$$(F \lor T \lor F) \land (F \lor T \lor T)$$

3 Graph into Boolean Expression

This problem can also be solved in the other direction. We can start with a graph, then build a boolean expression that will tell us how to color the graph. In this section, we will find a coloring using logic.

We will start with the below graph.



Question 7B [9pts]

(a) (2 points) This graph has four nodes. Each one needs to have a color. We will use R for red, Y for yellow, and B for blue.

Node 1 must be either red, yellow or blue. We can write this as a boolean expression $R_1 \vee Y_1 \vee B_1$. The variable R_1 is true if Node 1 is red.

Write a boolean expression that says Node 2 has at least one color. It might have more than one color, but we will resolve that in later questions.

R2 v Y2 v B2

(b) (2 points) Write a boolean expressions that says Node 3 has at least one color.

R3 v Y3 v B3

(c) (1 point) A node cannot have two colors, but our previous statements allowed that. We can add additional boolean statements to restrict the color choices further.

We can say "if node 1 is red then it is not blue" by writing $\neg R_1 \vee \neg B_1$.

What is $R_1 \implies \neg B_1$ in CNF?

¬(R1 ∧ B1)

(d) (2 points) Write "if node 1 is blue then it is not red" as a boolean expression.

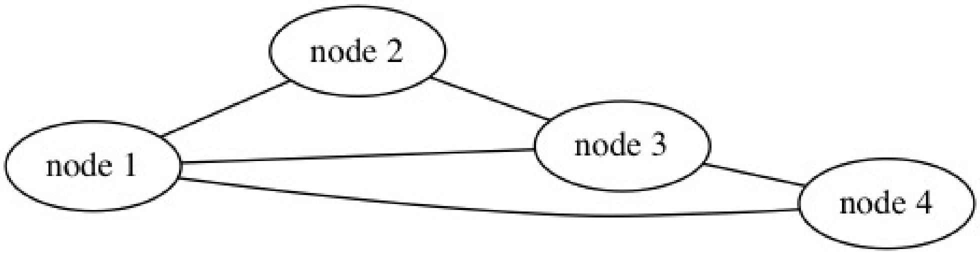
B1 ⇒ ¬R1

(e) (2 points) Write "if node 1 is blue then it is not red" in CNF.

¬(R1 ∧ B1)

Question 8 : 7 points

The graph also has edges. The edges further restrict what colors each node can be.



We can use additional implies expressions to handle these cases.

There is an edges from Node 1 to Node 3. This means if Node 3 is red, then Node 1 can't be red.

The following implies cover all possible restrictions related to the edge connecting Node 1 and Node 3.

$$
\begin{aligned}
R_1 &\implies \neg R_3 \\
B_1 &\implies \neg B_3 \\
Y_1 &\implies \neg Y_3 \\
R_3 &\implies \neg R_1 \\
B_3 &\implies \neg B_1 \\
Y_3 &\implies \neg Y_1
\end{aligned}
$$

(a) (3 points) Some of the conditional statements are redundant. Only three of these statements are actually required. Which three?

R1 ⇒ ¬R3
B1 ⇒ ¬B3
Y1 ⇒ ¬Y3

(b) (4 points) Prove that $(R_1 \implies \neg R_3) \iff (R_3 \implies \neg R_1)$ by drawing a Truth Table.

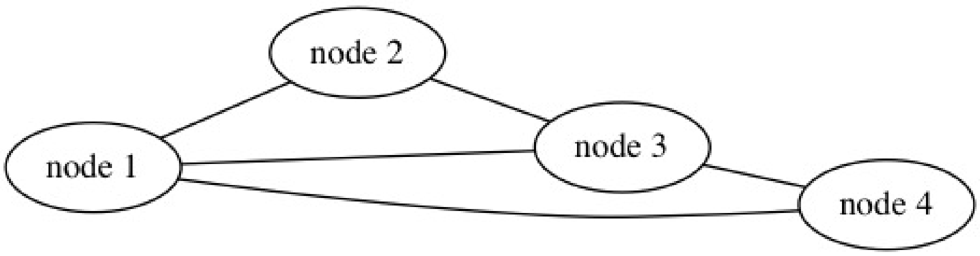
R1	R3	((R1 → ¬R3) ↔ (R3 → ¬R1))
F	F	T
F	T	T
T	F	T
T	T	T

Question 9 : 10 points

Using the concepts described in the previous pages, we can come up with a general algorithm for converting a graph color problem into a boolean expression.

1. For every node i in the graph, make an expression $R_i \lor B_i \lor Y_i$.
2. For every node i in the graph, make expressions $\neg R_i \lor \neg B_i$, $\neg Y_i \lor \neg B_i$, and $\neg R_i \lor \neg Y_i$.
3. For every edge connecting node i to node j, make expressions $\neg R_i \lor \neg R_j$, $\neg B_i \lor \neg B_j$, and $\neg Y_i \lor \neg Y_j$.
4. Create an single expression for the graph by AND-ing all the OR expressions we have created.

Review the following graph.



(a) (4 points) What are the 4 boolean expressions that follow from Step 1 of the Algorithm?

(R1 ∨ B1 ∨ Y1)
(R2 ∨ B2 ∨ Y2)
(R3 ∨ B3 ∨ Y3)
(R4 ∨ B4 ∨ Y4)

(b) (3 points) What are the 3 expressions that follow from Step 2 of the Algorithm for **Node 2**?

¬R2 ∨ ¬B2
¬Y2 ∨ ¬B2
¬R2 ∨ ¬Y2

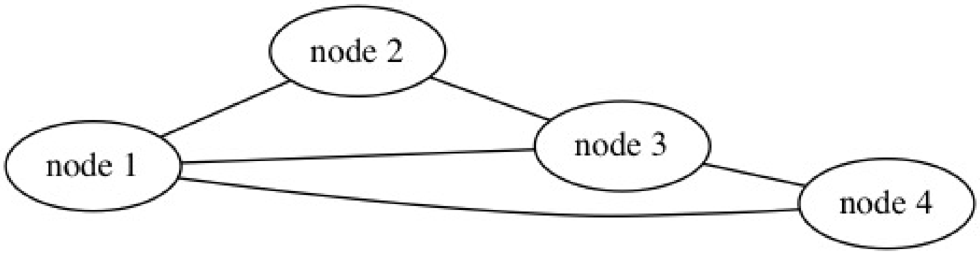
(c) (3 points) What are the 3 boolean that follow from Step 3 of the Algorithm for the **edge** connecting **Node 2** and **Node 3**?

¬R2 ∨ ¬R3
¬B2 ∨ ¬B3
¬Y2 ∨ ¬Y3

Question 10 : 14 points

DIMACS is the format you saw for MiniSAT.

In question 9, you came up with formulas for some of the constrains of the below graph.



(a) (10 points) Use the patterns you found in Question 9 to give all the constraints on the graph. Write a DIMACS format expression for this graph.

Use the SAT Solver linked in the course page navigation panel

Submit a Screenshot of your code running in MiniSAT.

You should have a comment block at the start of your code defining all your variables, as well as commented headings on each section of code explaining what those lines represent.
It will likely take you multiple screenshots to show all of your DIMACS data, since it will require over 30 clauses

```
c R1    B1    Y1    R2    B2    Y2    R3    B3    Y3    R4    B4    Y4
c  1     2    3      4     5    6     7     8     9     10    11    12
p cnf 12 30

1     2     3     0
4     5     6     0
7     8     9     0
10     11     12     0
-4     -5     0
-6     -5     0
-4     -6     0
-1     -2     0
-3     -2     0
-1     -3     0
-7     -8     0
-9     -8     0
-7     -9     0
-10     -11     0
-12     -11     0
-10     -12     0
-1     -4     0
-2     -6     0
-3     -7     0
-4     -7     0
-5     -9     0
```

```
-6     -10     0
-7     -10     0
-8     -11     0
-9     -12     0
-1     -7     0
-2     -8     0
-3     -9     0
-1     -10     0
-2     -11     0
-3     -12     0
```

Verdict: *SATISFIABLE*

```
SATISFIABLE

-1 -2 3 4 -5 -6 -7 8 -9 10 -11 -12 0
```

(b) (4 points) Use MiniSat to find an assignment of variables that solves the 3-color problem for the given graph. What color should each node be?

SATISFIABLE

-1 -2 3 4 -5 -6 -7 8 -9 10 -11 -12 0

Node 1: Yellow
Node 2: Red
Node 3: Blue
Node 4: Red