

# Scheduling, Map Coloring, and Graph Coloring

# Scheduling via Graph Coloring:

## Final Exam Example

Suppose want to schedule some final exams for CS courses with following course numbers:

1007, 3137, 3157, 3203, 3261, 4115, 4118, 4156

Suppose also that there are no students in common taking the following pairs of courses:

1007-3137

1007-3157, 3137-3157

1007-3203

1007-3261, 3137-3261, 3203-3261

1007-4115, 3137-4115, 3203-4115, 3261-4115

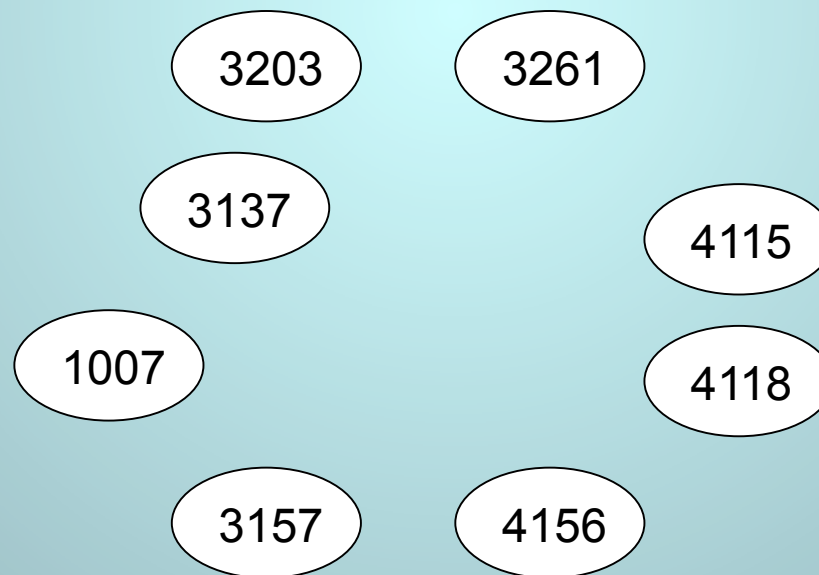
1007-4118, 3137-4118

1007-4156, 3137-4156, 3157-4156

How many exam slots are necessary to schedule exams?

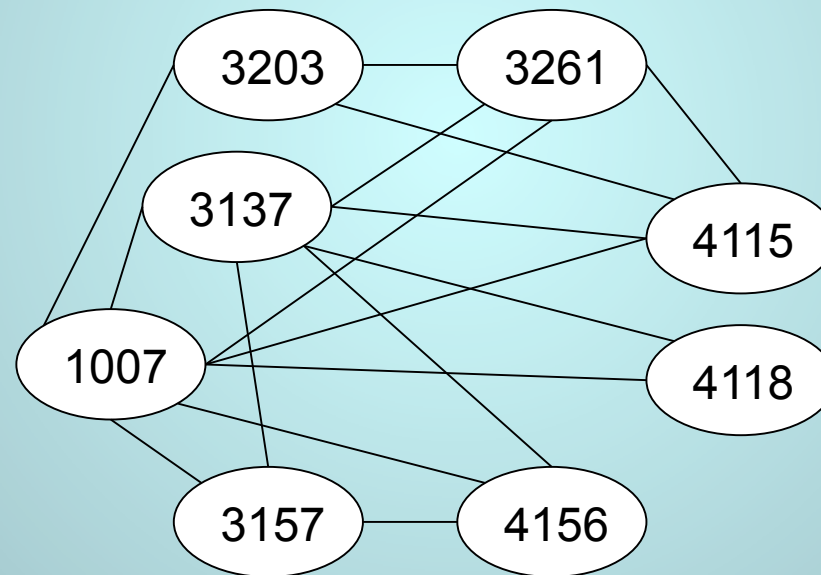
# Graph Coloring and Scheduling

- Convert problem into a **graph coloring** problem.
- Courses are represented by vertices.
- Two vertices are connected with an edge if the corresponding courses have a student in common.



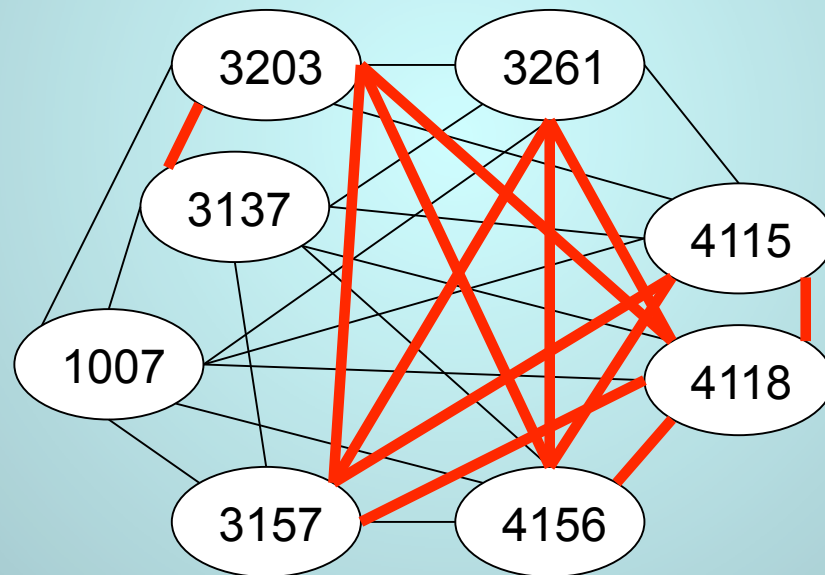
# Graph Coloring and Scheduling

One way to do this is to put edges down where students **mutually excluded**...



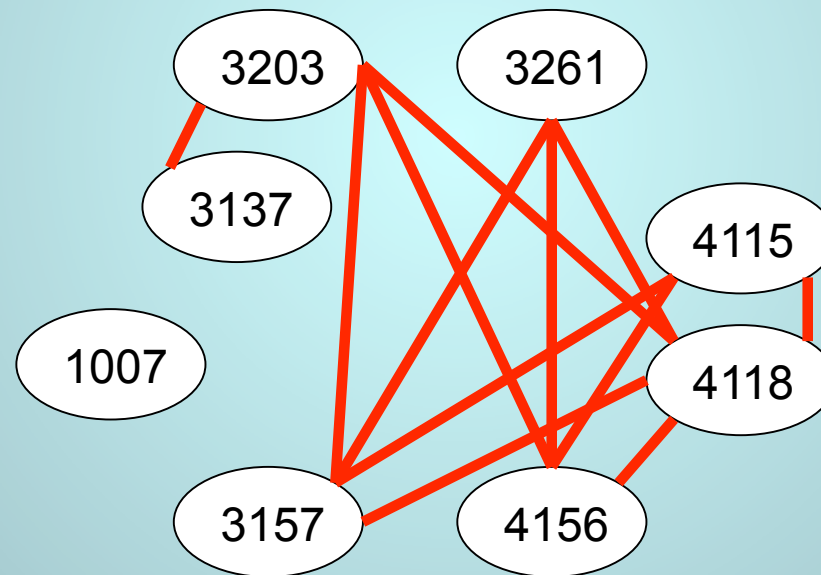
# Graph Coloring and Scheduling

...and then compute the **complementary** graph:



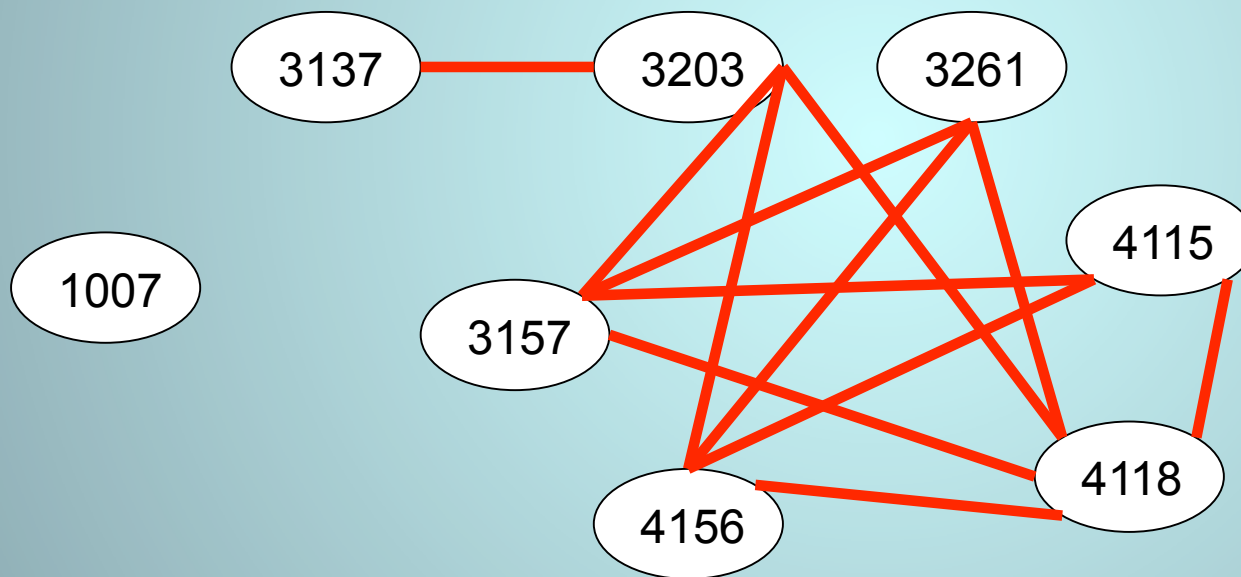
# Graph Coloring and Scheduling

...and then compute the **complementary** graph:



# Graph Coloring and Scheduling

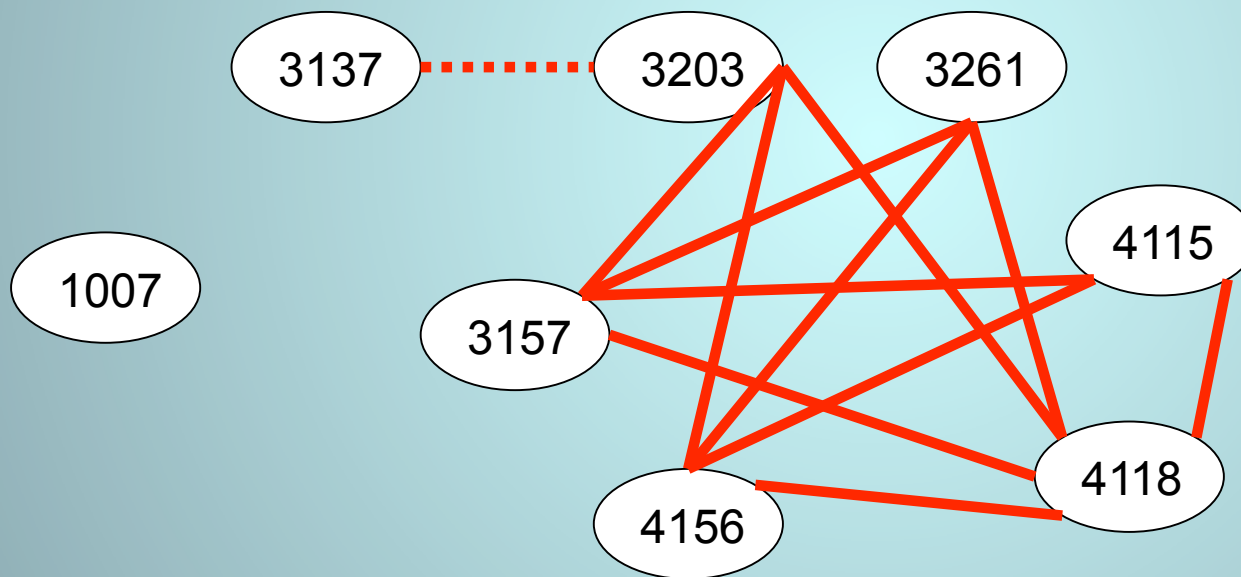
Redraw the graph for convenience:





# Graph Coloring and Scheduling

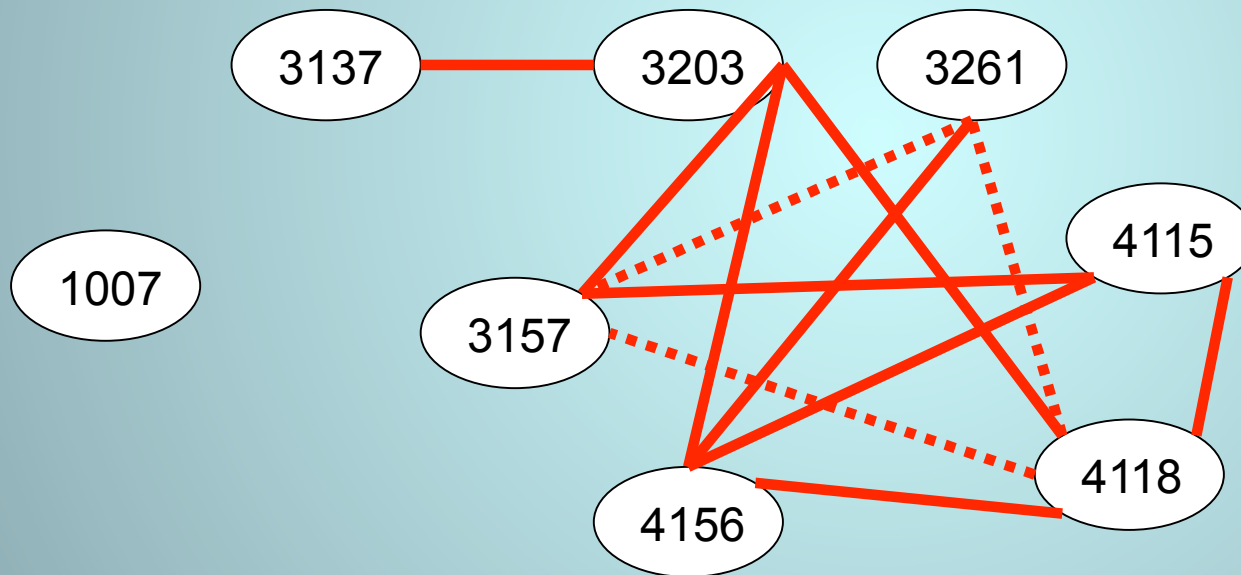
The graph is obviously not 1-colorable because there exist edges.





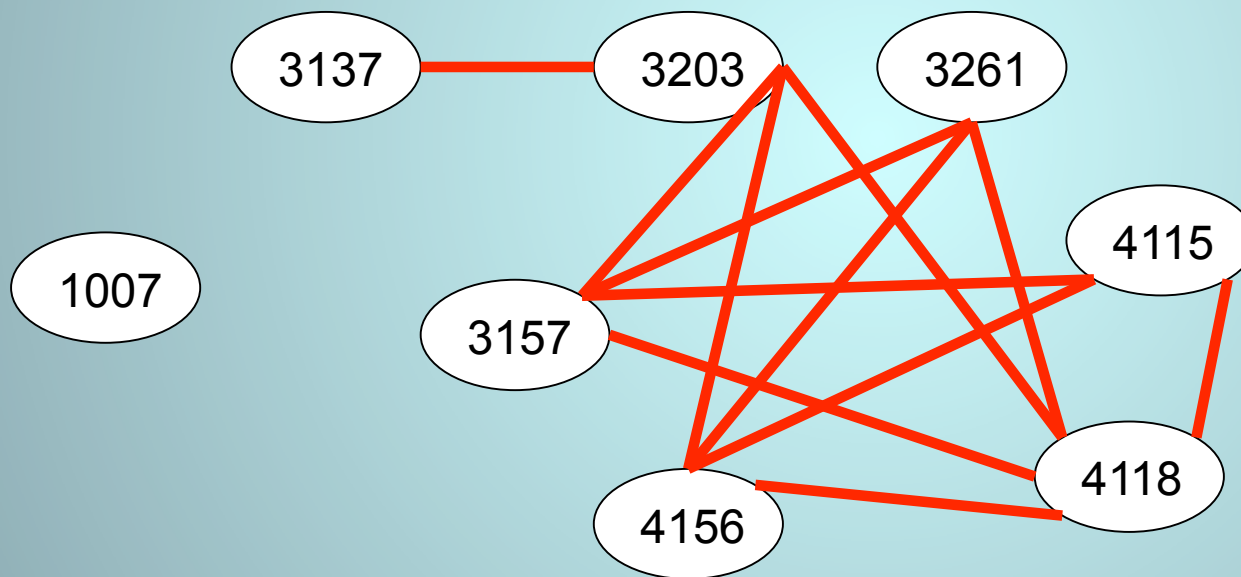
# Graph Coloring and Scheduling

The graph is not 2-colorable because there exist triangles.



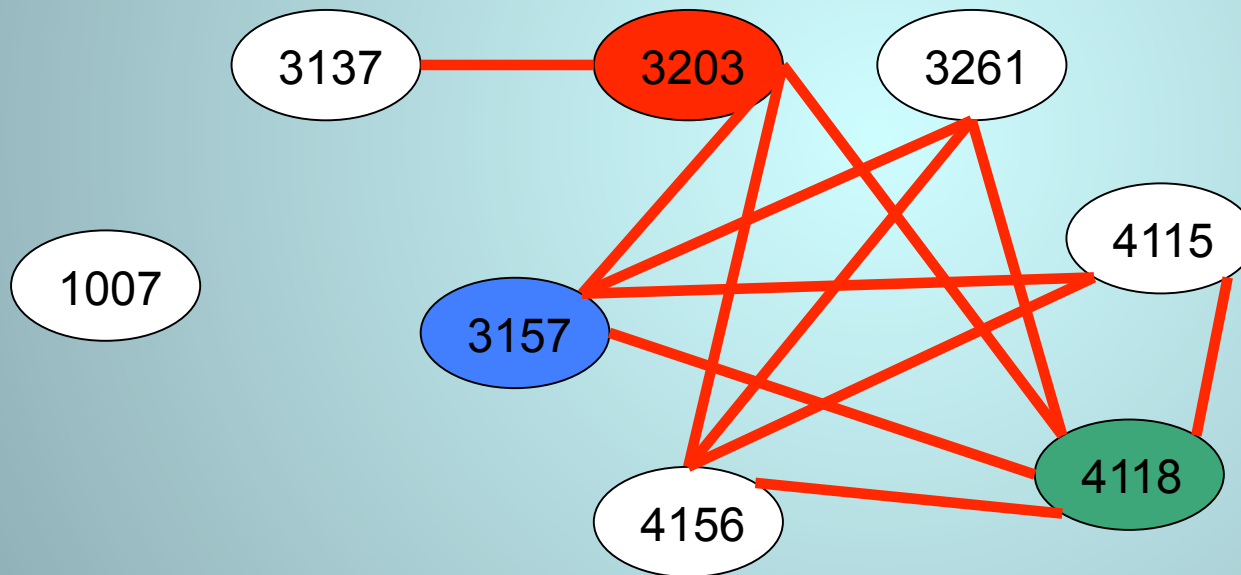
# Graph Coloring and Scheduling

Is it 3-colorable? Try to color by Red, Green, Blue.



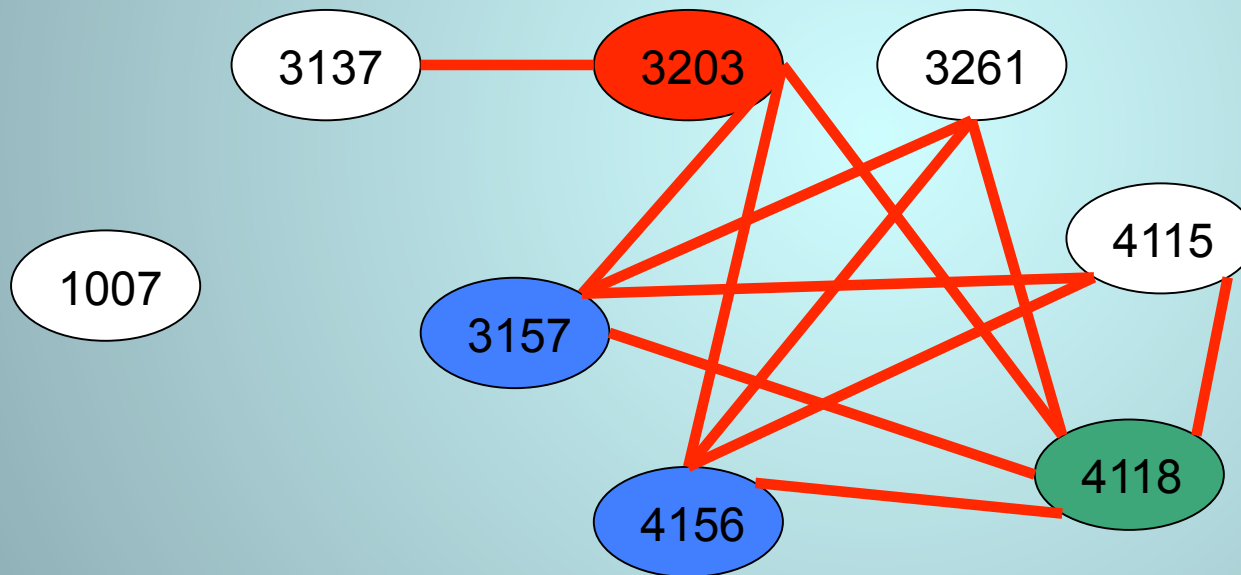
# Graph Coloring and Scheduling

Pick a triangle and color the vertices **3203-Red**, **3157-Blue** and **4118-Green**.



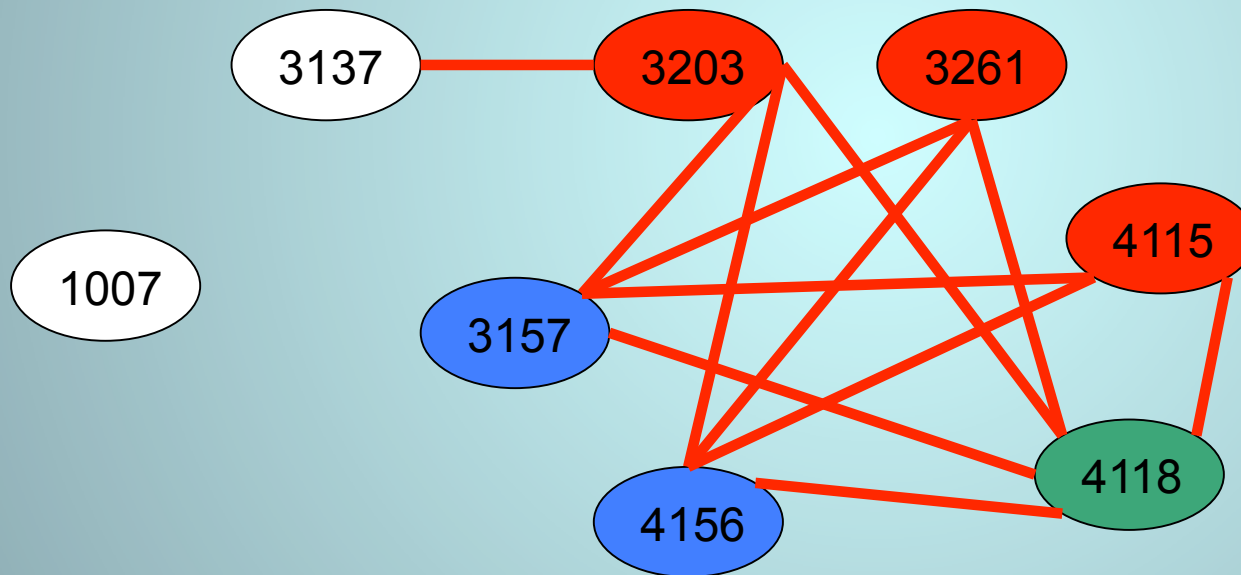
# Graph Coloring and Scheduling

So 4156 must be Blue:



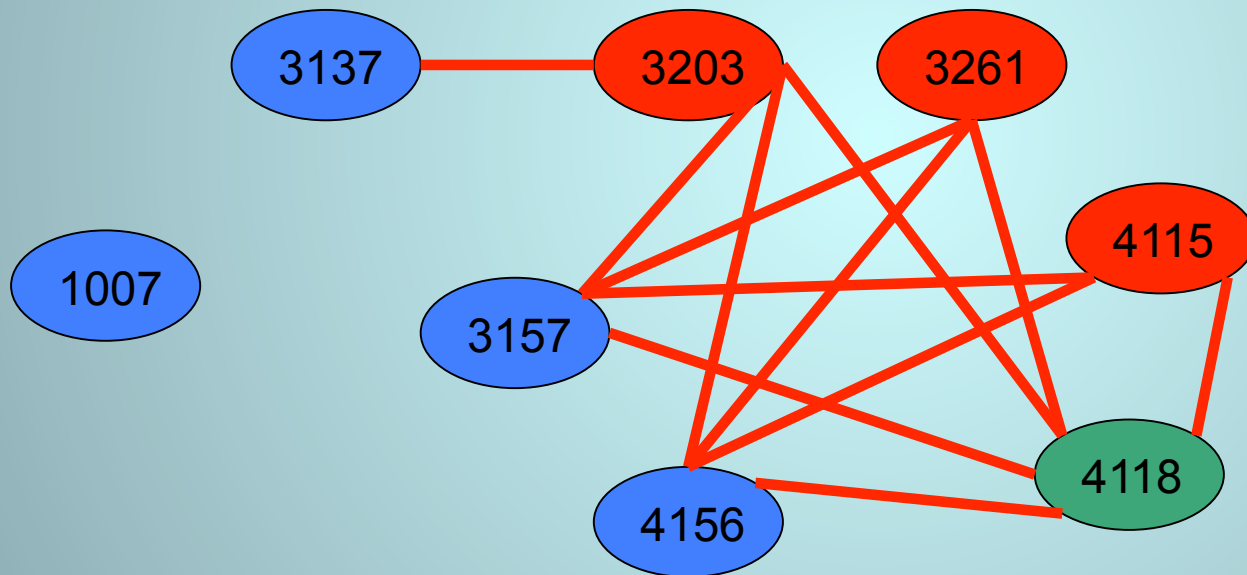
# Graph Coloring and Scheduling

So 3261 and 4115 must be Red.



# Graph Coloring and Scheduling

3137 and 1007 easy to color – pick Blue.



# Graph Coloring and Scheduling

Therefore we need 3 exam slots:

