Graph Coloring Problem

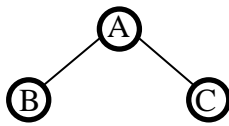A graph coloring algorithm works on an undirected graph G.

Each node in G has a property called "color".

A graph coloring problem is to color every node in G with minimal type of color so that every edge in G will connect two nodes with different color.

For example, if we have three nodes: A, B, C., and we have two edges A--B, and A--C.

The graph will look like this:

  B--A--C

We can use minimal two colors to color the graph:

Color 1: A

Color 2: B, C

Note that, all edges in the graph connects two nodes with different color.

The C++ program will need to solve the graph coloring problem.

You can use boost graph library to represent the graph internally.

The input of the graph will be an XML file, which look like this:

```xml
<graph>
      <nodes>
            <node name="A"/>
            <node name="B"/>
            <node name="C"/>
      </nodes>
      <edges>
            <edge src="A" dst="B" />
            <edge src="A" dst="C" />
      </edges>

</graph>
```

The program need to solve the problem by find out the minimal number of colors and a legal color assignment.

In your C++ program, you can use the greedy coloring algorithm or any other algorithm to solve the problem, but you cannot directly call the build-in graph coloring function in boost graph library.

However, you can use this built in function to check the validity and optimality of your own solution.