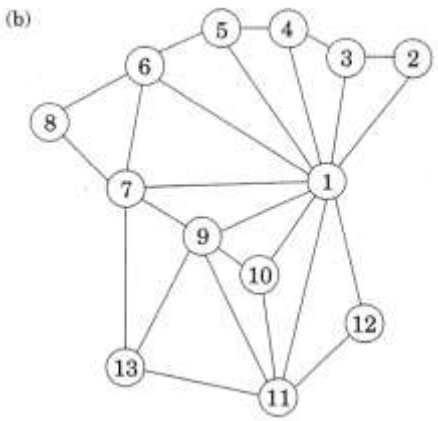
PROJECT 8

GRAPH ALGORITHM AND BREADTH FIRST SEARCH TREE

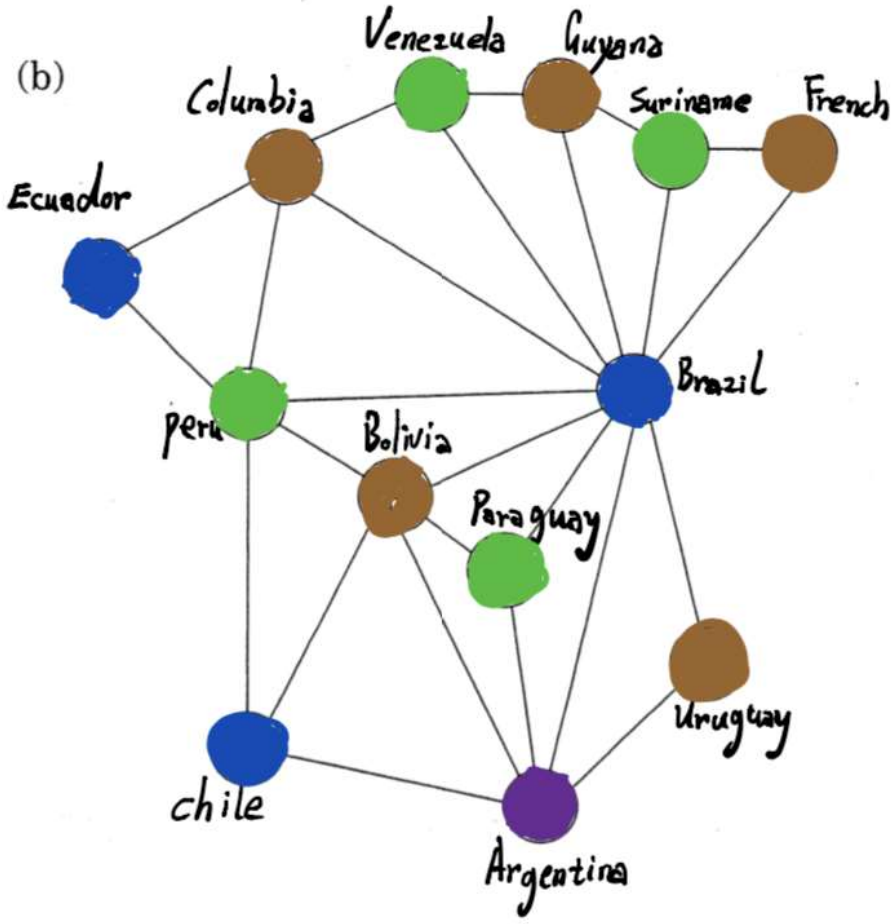
In graph theory, breadth-first search (BFS) is a strategy for searching in a graph when search is limited to essentially two operations:

1. Visit and inspect a node (vertex) of a graph;
2. Gain access to visit the nodes that neighbor to the currently visited node.

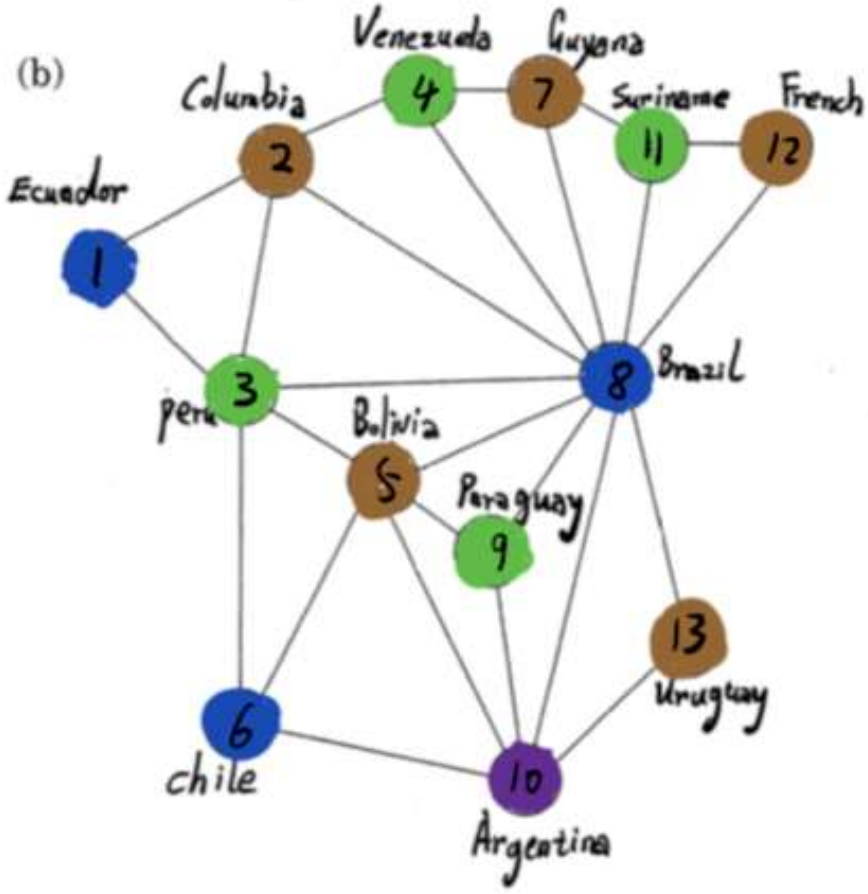
The BFS begins at a root node and inspects all the neighboring nodes. Then for each of those neighbor nodes in turn, it inspects their neighbor nodes which were unvisited, and so on.



a) Using the graph algorithm discussed in the class, implement a program in any language you desire to do Map Coloring Problem for the attached map of South America:
   (1) The neighboring countries must use different colors.
   (2) Convert the map coloring problem to a graph diagram from the attached map of South America.
   (**Draw the graph diagram.** You may use pen/pencil to draw this in a very nice drawing if utility software is not available.)



(3) **Display** the adjacency list representation of the graph from (a.2)



Adjacent list:

1 → 2 → 3 /

2 → 1 → 3 → 4 → 8 /

3 → 1 → 2 → 8 → 6 /

4 → 2 → 7 → 8 /

5 → 3 → 8 → 9 → 10 → 6 /

6 → 3 → 5 → 10 /

7 → 4 → 11 → 8 /

8 → 12 → 11 → 7 → 4 → 2 → 3 → 5 → 9 → 10 → 13 /

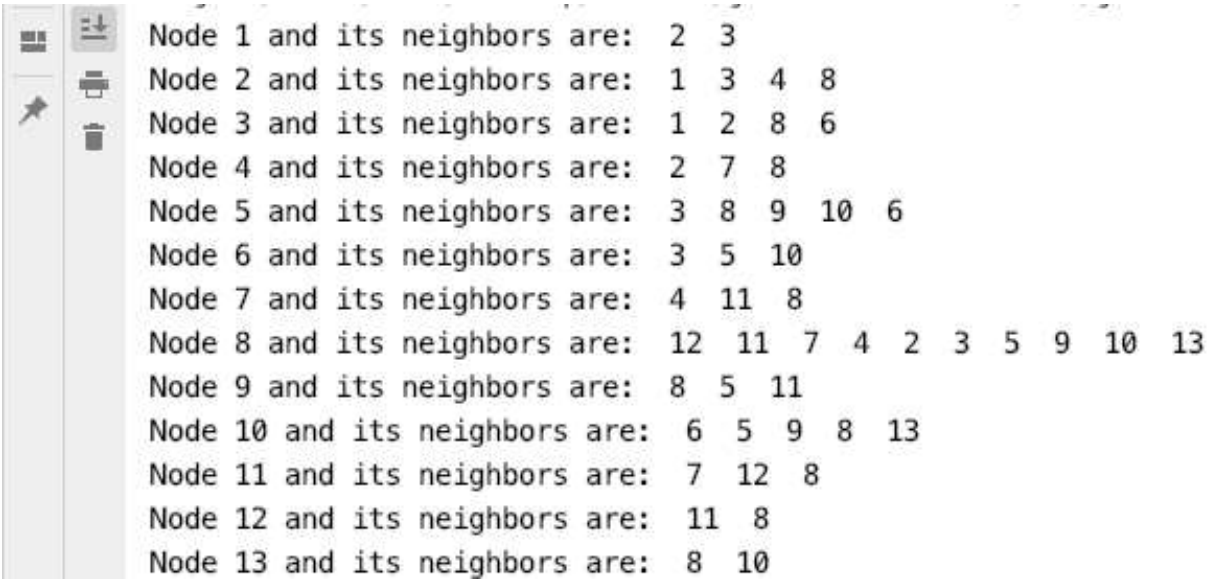
9 → 8 → 5 → 10 /

10 → 6 → 5 → 9 → 8 → 13 /

11 → 7 → 12 → 8 /

12 → 11 → 8 /

13 → 8 → 10 /



```
Node 1 and its neighbors are:  2  3
Node 2 and its neighbors are:  1  3  4  8
Node 3 and its neighbors are:  1  2  8  6
Node 4 and its neighbors are:  2  7  8
Node 5 and its neighbors are:  3  8  9  10  6
Node 6 and its neighbors are:  3  5  10
Node 7 and its neighbors are:  4  11  8
Node 8 and its neighbors are:  12  11  7  4  2  3  5  9  10  13
Node 9 and its neighbors are:  8  5  11
Node 10 and its neighbors are:  6  5  9  8  13
Node 11 and its neighbors are:  7  12  8
Node 12 and its neighbors are:  11  8
Node 13 and its neighbors are:  8  10
```

(4) **Display** the color used for each country and the colors for all countries.
(Find an algorithm BFS tree to use the minimum number of colors.)

```
1
Vertex Ecuador ---->  Color Blue
2
Vertex Columbia ---->  Color Brown
3
Vertex Peru ---->  Color Green
4
Vertex Venezuela ---->  Color Blue
5
Vertex Bolivia ---->  Color Blue
6
Vertex Chile ---->  Color Brown
7
Vertex Guyana ---->  Color Brown
8
Vertex Brazil ---->  Color Lavander
9
Vertex Paraguay ---->  Color Brown
10
Vertex Argetina ---->  Color Green
11
Vertex Suriname ---->  Color Blue
12
Vertex French ---->  Color Brown
13
Vertex Urugway ---->  Color Blue
```

(5) The colors used should be in the following order:
{Blue, Brown, Green, Lavender, Orange, Pink, Red, Yellow, Violet, Gold, Gray, Indigo, Silver}

```
private String[] colors = new String[]{"Blue", "Brown", "Green", "Lavander", "Orange", "Pink", "Red", "Yellow", "Violet", "Gold", "Gray", "Indigo", "Silver"};
```

```
The color sequence:
BlueBrownGreenLavanderOrangePinkRedYellowVioletGoldGrayIndigoSilver1
```