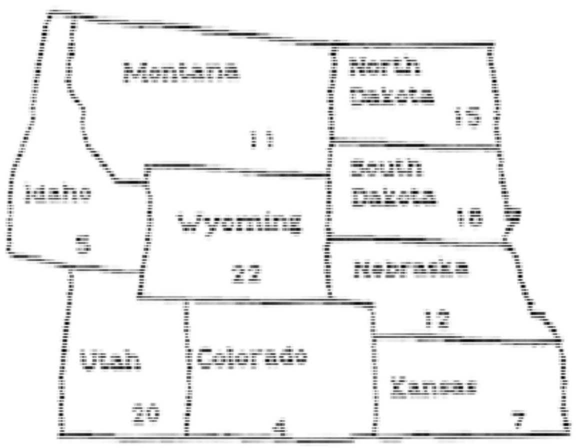
Midterm In-Class Programming Exercise

Overview:

Given a map of territories and a collection of different colors, determine if each of the territories can be colored using a different color so that no two neighboring territories are colored with the same color.

Example:



We can solve this using a *recursive backtracking* algorithm. In this case, Idaho and Wyoming cannot be the same color, nether can Utah and Wyoming.

Representation:

To solve this problem, use the following representation, as it will help you develop your solution.

Suppose there are `n` territories. We will use a 2D array called *neighbors* that is `n x n` in size. If territory `i` is a neighbor of territory `j`, then `neighbors[i][j] = neighbors[j][i] = true`. Otherwise, `neighbors[i][j] = neighbors[j][i] = false`.

Use an array to represent all the possible colors you can use. Example, `allColors = {"Red", "Blue", "Green", "Yellow"}`.

Use an array called `territoryColors` of size `n`, where `territoryColors[i] = the index in allColors that is used for territory i`.

Inputs:

Imagine that other programmers are using your code. To solve the problem, they will call a method you write called `int[] generateColorMap(String[] allColors, boolean[][] neighbors)`. If there is a way to color all the territories using the colors provided, return an array that specifies the color index for that territory. If no solution exists, return `null`.

Write another method called `void displaySolution(String[] allColors, int[] territoryColors)`. Given the output from `generateColorMap ()`, if a solution exists, this method will print the `String` color name for each territory in the list.

Sample Runs:

Run 1 (Possible Solution)	**Run 2 (No Solution)**

boolean neighbors[][] = {
{false, true, true, true},
{true, false, true, false},
{true, true, false, true},
{true, false, true, false},
};
 	
String[] availableColors = {"RED", "GREEN", "BLUE"};
 	
int[] territoryColors = generateColorMap(availableColors, neighbors);
 	
if (territoryColors != null)
{
displaySolution(availableColors, territoryColors);
}
else
{
System.out.println("No solution exists");
}
	boolean neighbors[][] = {
	{false, true, true, true},
	{true, false, true, false},
	{true, true, false, true},
	{true, false, true, false},
	};
	
	String[] availableColors = {"RED", "GREEN"};
	
	int[] territoryColors = generateColorMap(availableColors, neighbors);
	
	if (territoryColors != null)
	{
	displaySolution(availableColors, territoryColors);
	}
	else
	{
	System.out.println("No solution exists");
	}

Output:	**Output:**

`Territory 0 = RED`
`Territory 1 = GREEN`
`Territory 2 = BLUE`
`Territory 3 = GREEN`
	`No solution exists`