CSCE 451 Lab3

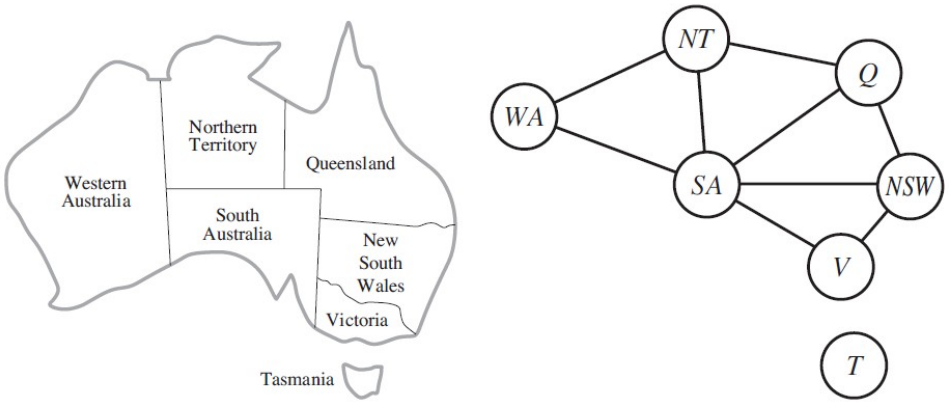
Due July 21 at 6:00pm

A simple graph coloring with DFS and degree heuristics

In this lab we implement and investigate, a simple DFS search program that colors the map graph from our text book.

Task 1: What we need?

We intent to implement a program that colors the map below. This map is taken from the text book and by now we should be familiar with it. The ultimate goal of this assignment is to develop a program that colors the map such that for any neighboring states (nodes) we have different color assigned.



Download DFSGraphColoringAssignment.zip. Run the code and let's discuss it in the class. Make sure you understand the code.
The code uses the graph class that we wrote previously in assignment1. A new class is added for DFS recursive search traversal. You can change the input examples to any graph you like for testing the program.

Task 2: Coloring algorithm
For coloring the graph. We are going to use the simplest method proposed by the text book – **degree heuristic** as starting point**.** For the coloring process we will a simple use **Maintaining Arc Consistency** (MAC) algorithm.

Degree heuristic: We need to have a function to return the most connected node. This node will be the starting point.

MAC: After assigning a color to the start node (the highest degree), that color is removed from list of colors of its children.

Task 3: DFS for coloring

1- A function that returns the most connected node. This function initializes the DFS algorithm with the most connected node.

2- All graph node is be initialized with available list of colors.
3- A color will be assigned to a node from list of available colors for that node.
4- That color in step 3 will be removed from the color's list of each child. See the figure below.

	WA	*NT*	*Q*	*NSW*	*V*	*SA*	*T*
Initial domains	R G B	R G B	R G B	R G B	R G B	R G B	R G B
After *WA=red*	(R)	G B	R G B	R G B	R G B	G B	R G B
After *Q=green*	(R)	B	(G)	R B	R G B	B	R G B
After *V=blue*	(R)	B	(G)	R	(B)		R G B

Task 4: Experiments:
The output example

```
C:\Users\arjan\Desktop\Course\Summer 2020\CECS451\LabActivity\Lab3\DFSGraphColoring>main.py
2 --------> Red
0 --------> Blue
1 --------> Green
3 --------> Blue
4 --------> Green
5 --------> Blue
```

A- What is the candidate node for the starting node? Why?
B- What will be the output if we change the start node to other nodes?
C- What is the out put of the program if we use 2 colors instead of three? Test your guess by removing a color from the list.
D- What is the out put of the program if we use 4 colors instead of three? Test your guess by adding a new color to then list.
E- Can you come up with a graph that this algorithm doesn't work correctly? Why?
    a. Make your guess as a graph input file, and run the code and observe the result.