CMPT 310 - Artificial Intelligence Survey

Assignment 2

Due date: February 26, 2018 J.P. Delgrande
10 marks February 6, 2018

Important Note: Students must work individually on this, and other CMPT 310, assignments. You may not discuss the specific questions in this assignment, nor their solutions with any other student. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the general concepts involved in the questions in the context of completely different problems. If you are in doubt as to what constitutes acceptable discussion, please ask!

Write a program in Python that solves instances of map colouring problems. A map colouring problem is made up of a set of vertices (representing regions), edges (representing adjacency), and a set of colours, or values for variables represented by vertices. A solution is an assignment of colours to vertices such that every vertex has a colour and no two adjacent vertices have the same colour.

Assume for n vertices that the vertices are numbered 1 through n; similarly m colours are numbered 1 through m. The graph will be described by a list of lists, where for a list element $(v_1\ v_2 \ldots v_n)$ means that v_1 is connected to each of $v_2 \ldots v_n$. (That is, the graph is specified by an adjacency list.) For example

((1 2 3) (2 1 3) (3 1 2) (4 5) (5 4))

describes a graph with 5 vertices. You can assume that the graph is undirected and that the input is correct (i.e. don't bother with error checking). Note that since edges are undirected, each edge is given twice.

Please use the following conventions for input and output. Your main function will be called `solve` and will have arguments for the number of vertices, number of colours, and the graph, for example:

```
G = [[1,2,3], [2,1,3], [3,1,2], [4,5], [5,4]] # a list of lists
n = 5  # number of vertices
k = 3  # number of colours
def solve(n, k, G):
    # write your code here
    return solution
```

The value of `solution will be a list of tuples`. Each tuple represents an assignment of a colour to a vertex such that every vertex has a colour and no two adjacent vertices have the same colour. If it is not possible to find such an assignment with k colours, the solution will be an empty list. For example:

```
solution = solve(5, 3, G)
print(solution)  # [(1,1), (2,2), (3,3), (4,1), (5,2)]

solution = solve(5, 2, G)
print(solution)  # []
```

For the assignment, implement basic backtracking search along with the MRV, degree, and least constraining value heuristic. In your documentation, make sure that you describe your program at a high level and discuss any interesting aspects of your program. Test data is in a separate file, asking for a 4-colouring of 10 countries. Run your program on this data, as well as trying for a 3-colouring (which will fail). As well, optionally, test your program with and without using the heuristics, and report on the improvement (or lack of improvement) obtained.