Programming Assignment 2 – AI I – Spring 2018

The purpose of this assignment is to practice on the search algorithms that you have implemented in the first assignment [breadth first search (BFS), depth first search (DFS), iterative deepening search (IDE), and A* search].

The setting: we will use the graph coloring problem.

1. *Preparation*: Assume that we are given a text file with the following information

   The first line of the file is of the form: N E C – it has three integers N, E, and C separated by one (or more) space. N, E, and C are the number of nodes, the number of edges, and the number of colors, respectively.

   For example: the first line is

   ```
   7  6  3
   ```

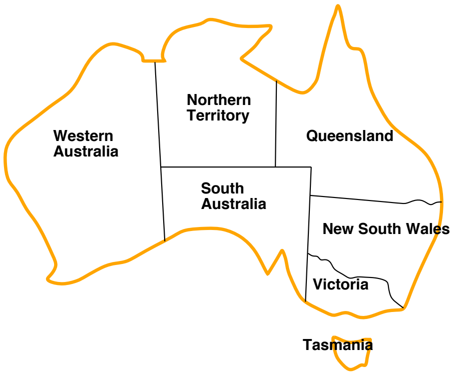
   and it says that the graph has 7 nodes and 6 edges; and we want to color it with three colors.

   We will assume that the nodes of the graph are numbered from 1 to N and the colors are numbers from 1 to C. In the example, we have the nodes are 1, 2, …, 7 and the color are 1, 2, 3.

   The following lines of the file are of the form: E1 E2, i.e., it contains two numbers E1 and E2; both are in the range between 1 and N. Each represents an edge of the graph.

   For example, the file with the following lines (on the left of the map):

   ```
   7  6 3
   1 2
   1 3
   2 3
   2 4
   3 4
   3 5
   3 6
   4 5
   5 6
   ```



   This graph represents the Australia map to the right if you assign the number 1, 2, 3, 4, 5, 6, and 7 to WA, NT, SA, Q, NSW, V, and T, respectively. Note that the edges are bi-directional.

2. *Implementation*: You should use your implementation for the first assignment to work on this one. Use the search algorithm to solve the graph coloring problem. For the A* algorithm, invent a heuristic and implement it.

   Your program must take a file name as an input, print out the solution to the problem, and report the number of states in the fringe and the number of states that have been expanded at the end of the search. It should be able to run on a terminal via command line.

3. *Submission*: please upload the following in a **single zip file** that contains the following:

   a. All of the codes needed to run your program.
   b. A README file for compiling/running your program.
   c. A set of at least five test problems; for each problem, the graph must have at least 10 nodes (we will call this **the test set**). Make sure that there is some problem that has no solution.
   d. A report about your evaluation of different search strategies in solving the problem. The report can be done by doing the following:
      i. Run your program with the test set.
      ii. Report your observations about the behaviors of the program with different inputs (e.g., which one is faster, which one needs to explore more/less nodes, which parameters of the input affect the running time the most?) and present your hypothesis to answer the question "why does the program behave like that?"
   e. A print out of the code that you added to solve the problem (**only whatever you added to the old program!**)

Name your submission using the following convention: "Last_Name_ID" (last name followed by your ID.

Grading (100 for full marks): Test set and report: 30 points and the program 70 points.

- Test set: 10 points.
- Satisfactory report: 20 points.
- README file: 5 points.
- Program:
  - Program cannot be compiled: a total of 10 point.
  - Program does not give correct answer for some inputs: total 30 points max!
  - Program gives correct answer for all problems: total 65 points.

10 bonus points for program that can solve the problem for the US map (provided the input file that you use for this!)

Points will be deducted for not following instruction!

Deadline: 11:59pm, 2/18/2018.