Foundations of Intelligent Systems
Assignment 1 (Prof. Richard Zanibbi)

Due: Friday, Sept. 9, 6pm (Submit through MyCourses dropbox)

This assignment is out of 50 points.

Submission: Submit the two files described below through the MyCourses dropbox. **A penalty will be applied for submissions that do not follow these instructions.**

1. `a1.pdf`, with your written answers. `a1.pdf` must be typeset using a word processing program (e.g. LaTeX or Word). **Photographs or scans of handwritten work will be penalized.**

   For question 3, you may use any **graphics editing tool** you wish to draw the search tree - I suggest using the 'dot' program from GraphViz, which can produce trees from graph definitions in text files. `dot` is installed on the CS computer systems, and can be downloaded at `www.graphviz.org`.

2. `a1.zip`, containing your Python code, Romania map file, and test output file. Use Python 3.5 (command '`python3`' on the CS computer systems)

Questions

1. Course text Question 3.7, parts *a* and *b*

2. Course text Question 3.11

3. Course text Question 3.15

4. Assume that the Romania map in the text and discussed in lecture is changed, so that all roads are 1 mile long. Write a Python program that finds optimal paths between a given pair of cities using *Iterative Deepening Search (IDS)*. Your IDS will need to be a *graph search*. Because there are cycles in the map, revisiting cities can lead to an infinite loop.

   The file `A1_init_code.zip` available through MyCourses contains a program `SearchGraph.py`. The program reads a graph from a text file (for this assignment, use the provided file `romania`), and the names for the initial and goal city names. Your program will output the following:

   (a) During the execution of iterative deepening, print the search tree each time all of the search tree nodes at the **depth limit for the current iteration of IDS** have been visited.

(b) At the end of the search:
   i. **If a solution exists**, return the list of cities in the solution, beginning with the initial city and ending with the goal city.
   ii. **Otherwise**, return a list with the string 'FAIL.'

You need to modify the file `SearchGraph.py`, and provide a text file `tests.txt`, that shows the execution of your program for 3 pairs of cities:

- Arad to Bucharest,
- two additional city pairs of your choosing (*interesting* cases).

A bash shell script `test` has been provided, which you can modify to automate the execution of your test cases (see the **README** for details).

Submit `SearchGraph.py`, `tests.txt` and the `romania` map file in a single .zip file `a1.zip`.