

PSet 1

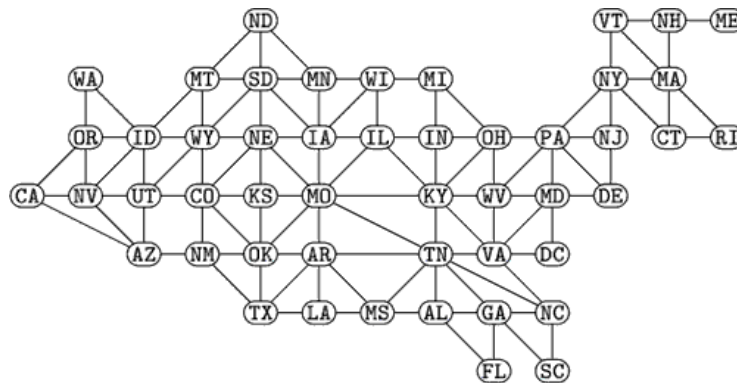
CS 7649 Robot Intelligence: Planning
Instructor: Matthew Gombolay

Instructions:

- You may work with one or more classmates on this assignment. However, all work must be your own, original work (i.e., no copy + pasting code). You must list all people you worked with and sources you used on the document you submit for your homework
- You must use Python version 3.6 to receive credit.**
- Zip all code and PDFs before submitting as a single, zipped folder.

Problem 1:

Consider the following graph (credit Knuth 2008, p. 15):



Write six Python scripts implementing depth-first search (DFS), breadth-first search (BFS), and Iterative Deepening Search (IDS), each with and without a visited list. Use these algorithms to solve for a simple path from $S \rightarrow G$, where S is Washington (WA) and G is Georgia (GA). Ties will be broken in alphabetical order. Augment your implementation to report the computational time, # of paths popped from the queue, maximum queue size, and length of returned path. You may manually define the graph in python or grab a definition from online. **Submit a PDF of this page having filled in these rows and columns:**

| | Visited List | Time (s) | # Paths Popped from Queue | Max Queue Size | Returned Path's Length/Cost |
|-----|--------------|----------|---------------------------|----------------|-----------------------------|
| DFS | No | | | | |
| | Yes | | | | |
| BFS | No | | | | |
| | Yes | | | | |
| IDS | No | | | | |
| | Yes | | | | |

You should have one Python file for each type of search, and the code should be clearly documented. Files should be named “[METHOD]_[VISITED].py” (e.g., “dfs_yes.py”). Further, when the grader runs the python code, the script should print to the screen the relevant information reported in the table above. **Remember, if it does not run, you do not receive credit.**