CSC520 Spring 2018 - Assignment 1

January 2018
Due by midnight **February** 7^{th} **2018**.

This assignment includes both conceptual and code questions. It must be completed individually. You may not collaborate with other students, share code, or exchange partial answers. Questions involving answers or code must be emailed to the instructor or TAs directly or discussed during office hours. Your answers to the conceptual questions must be uploaded to Moodle as a pdf file titled "Assign1.pdf". Your sourcecode must be submitted as a self-contained zip. All code must be clear, readable, and well-commented.
Note: The code will be tested on the NCSU VCL system. You are advised to test your code there before submission.

1 Question 1 (20 points)

Suppose we have an agent who works for the Hunt Library. The agent is assigned a task of transferring a set of boxes one-by-one by lifting them from location A and placing them in location B inside the building. Sensors can tell it whether the agent is near its destination or not. The rooms have stationary obstacles whose locations are not initially known. If the agent bumps into an obstacle, it will drop the box it is carrying. Some boxes contain fragile contents which will break if dropped. The environment contains obstacle-free paths of various lengths. Answer the following questions about the agent:

(a) Define a PEAS specification for the agent.

(b) Is it sufficient for the agent to be simple reflex? Why or why not? And if not what level is necessary?

(c) Would the ability to move randomly help agent performance or not? Identify possible disadvantages to this sort of movement.

(d) Suggest one improvement to the agent design. Since every improvement carries drawbacks, what are the drawbacks to yours?

2 Question 2 (20 points)

1. For each of the following agents, state the characteristics of its environment in the given table:

Agent	Fully/Partially Observable	Single/ Multi-Agent	Deterministic/ Stochastic	Episodic/ Sequential	Static/ Dynamic	Discrete/ Continuous	Known/ Unknown
tax planner							
Search and rescue robots							
Document classifier							
fitness coach							
Packet router							

2. Select two of the agents and present a short (1-sentence) justification for each of the characteristics.

3 Question 3 (60 points)

Figure 1 shows a road map of the Unites States. Nodes represent cities, and the numbers on each edge indicate driving distances between the connected cities. The distances and lat/long coordinates for the cities may be found in the roads.txt file. In a language of your choice (Java, C, C++, Python), implement Depth-First Search, Breadth-First Search, and A* using the heuristic of straight-line distance between the cities.

- Your BFS and DFS search should follow alphabetical order, and the same for A* in case of ties.
- There is a complication in computing straight-line distance from longitude and latitude that arises because the earth is roughly a sphere, not a cylinder. In the provided file, the header comment indicating how the heuristic should work.
- Your code should be capable of tracking the expanded nodes as well as finding the solution path.
- The command that we use for testing your code -depending on the language you use- is:
  `java/./a.out/python SearchUSA searchtype srccity destcity`

- Name your source file and main function as `SearchUSA`. Searchtype can be either `bfs, dfs,` or `astar` and city names are the same as they appeared in the roads.txt file.
- Submit your code along with answers to the questions below. Your grade will be based on both your answers and the quality of your code.

Experiment by executing your code and answer the following questions:

(a) Consider the path from Minneapolis to Cleveland. Run your algorithms and show the nodes expanded by each one as well as the paths returned by both BFS and DFS.

(b) Is there a case where DFS outperforms BFS in terms of number of nodes expanded? If yes, what is the case? If not, explain why?

(c) Is there a case where DFS outperforms BFS in terms of number of nodes on the solution path? If yes, what is the case? If not, explain why?

(d) Are there any pairs of cities (A,B) for which the A* algorithm finds a different path from B to A than from A to B? If yes, what is the case? If not, explain why?

(e) Is there a pair of cities for which DFS outperforms A* in terms of the number of nodes expanded? If yes, what is the case? If not, explain why?

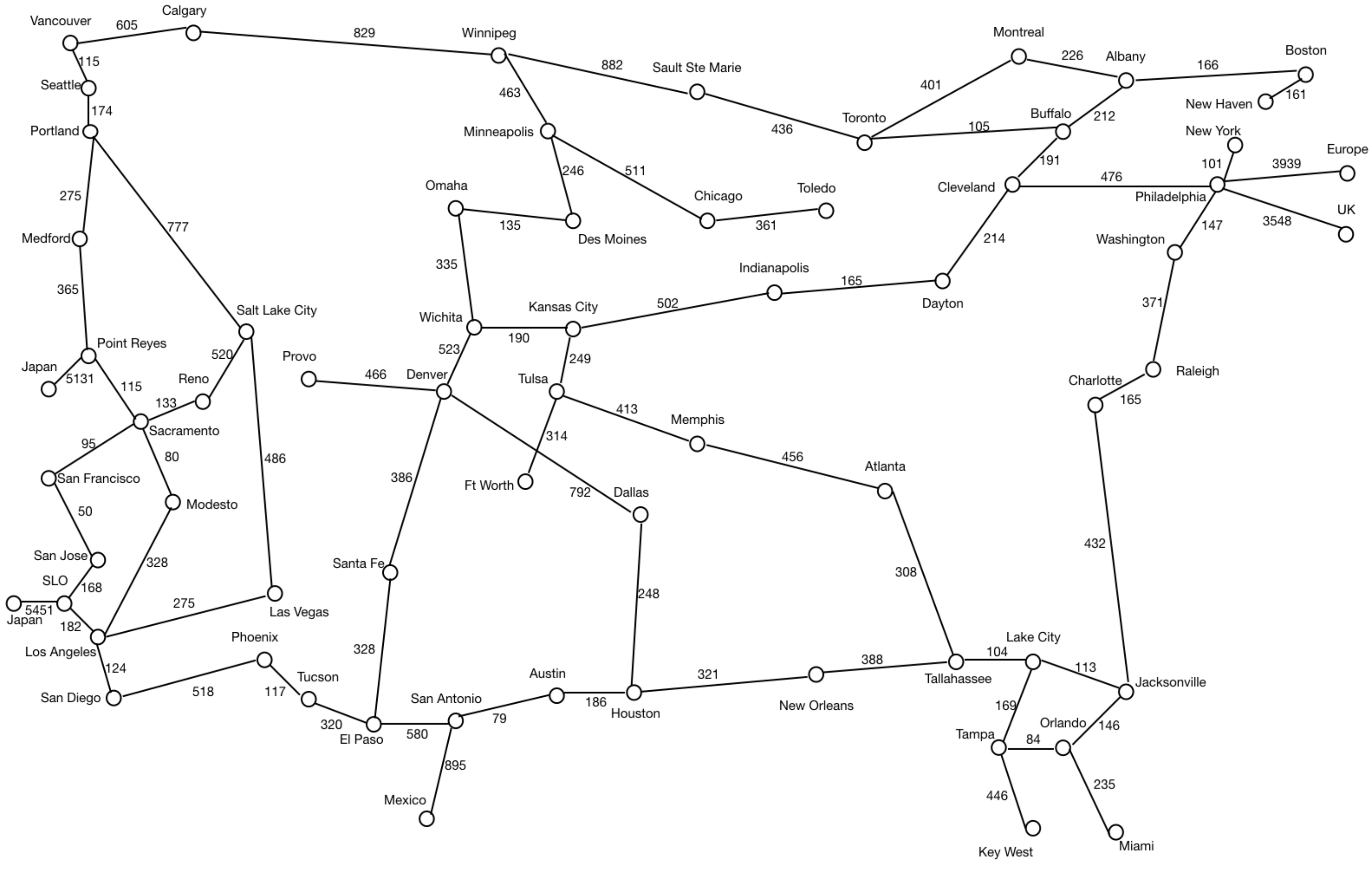

Figure 1: US roads map