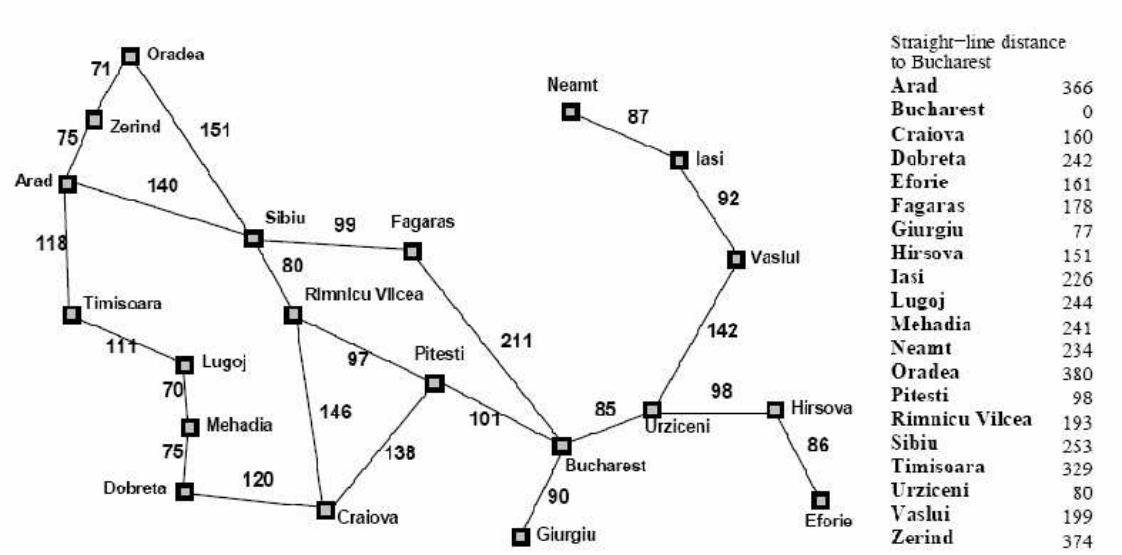
COS30019 - Introduction to Artificial Intelligence
Tutorial Problems Week 4

Task 1: Consider the problem of getting from Arad to Bucharest in Romania and assume the straight-line distance (SLD) heuristic will be used.

1. Give the part of the search space that is realized in memory and the order of node expansion for:
    a. Greedy search assuming that a list of states already visited is maintained.
    b. A* search assuming that a list of states already visited is maintained.
2. How would the above searches differ if the list of states already visited is NOT maintained?
3. How do the above searches perform for planing a trip from Iasi to Fagaras?



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

To provide you with the heuristics for Task 1.3, please use the following as the straight-line distance (SLD) between relevant cities and Fagaras:

Iasi	180
Neamt	160
Vasiul	200
Urziceni	175
Bucharest	178
Hirsova	255
Pitesti	130
Giurgiu	270
Eforie	320

Task 2:

1. Suppose we run a greedy search algorithm with $h(n) = -g(n)$. What sort of search will result?
2. Suppose we run an A* algorithm with $h(n) = 0$. What sort of search will result?
3. Explain why the set of states examined by A* is often a subset of those examined by breadth-first search.

Task 3: The missionaries and cannibals problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side, without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.

1. Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?
2. Is there a heuristic that would be useful for the missionaries and cannibals problem? The generalized missionaries and cannibals problem (*n* missionaries and *n* cannibals)?

Programming Task - Route problem with Heuristic Search

Last week you were required to design an uninformed search to find a route from one city to another. This week, you will develop an A* implementation, which uses a heuristic function to determine the path of greatest promise.

In Java, there exists a class called *PriorityQueue*. A priority queue accepts new objects that are added to it, but unlike a queue or a stack, it will place the new object in the queue based on its cost. To achieve this, the priority queue uses a Comparator. Read the API for both of these classes, and determine how you will implement a priority queue that orders elements appropriate for the lab. If you are using another language, you will need to look up the necessary classes that will achieve the same result.

Your main function should accept three arguments: the name of a file to read from, an origin city, and a destination city. Use the input file from last week to test your solution. The expected output is a list of cities traversed from the origin, all the way to the destination. Make sure to compare the result of your program with a hand worked solution to ensure that the correct path has been reached.

Extension:
Refer to the missionaries and cannibals problem discussed in the tutorial. Think of an appropriate heuristic, function, and implement a program that is able to discover the quickest way to get all people from one side of the river to the other. Think carefully about what moves are appropriate in a given state.