The University of Adelaide
School of Computer Science

Artificial Intelligence
Tutorial 1

*Note that some of the material required for this tutorial will not be covered until Friday of Week 2. This may mean that you will need to read around the subject in advance of the lectures to get the most out of the tutorial. In particular, reading the lecture notes for lectures 3 and 4, as well as chapters 3 and 4 of AIMA is recommended.*

**Question 1** (Question 3.8 of AIMA 2ed)
Consider a state space where the start space is 1 and the successor function for state $n$ returns two states: $2n$ and $2n+1$.

(a) Draw a portion of the state space up to state 15.

(b) Suppose the goal state is 11. List the order in which the nodes would be visited for:

1. Breadth-first
2. Depth-limited (to 3)
3. Iterative-deepening
4. Bi-directional search starts the search from the start state and end state, and stops as soon as the two search trees meet. Would bi-directional search work for this problem? If so describe how it would work.

**Question 2** (Question 3.2 of AIMA 3ed)
Your goal is to navigate a robot through a maze; see Figure 1. The robot enters the maze from the top left facing south. You can turn the robot to face north, east, south, or west. You can direct the robot to move forward a certain distance, although it will stop before hitting a wall.

(a) Formulate this problem. How large is the state space?

(b) In navigating a maze, the only place we need to turn is at the intersection of two or more corridors. Reformulate this problem using this observation. How large is the state space now?

(c) From each point in the maze, we can move in any of the four directions until we reach a turning point, and this is the only action we need to do. Reformulate the problem using these actions. Do we need to keep track of the robots orientation now?

(d) Define an appropriate path cost and heuristic function for this problem.

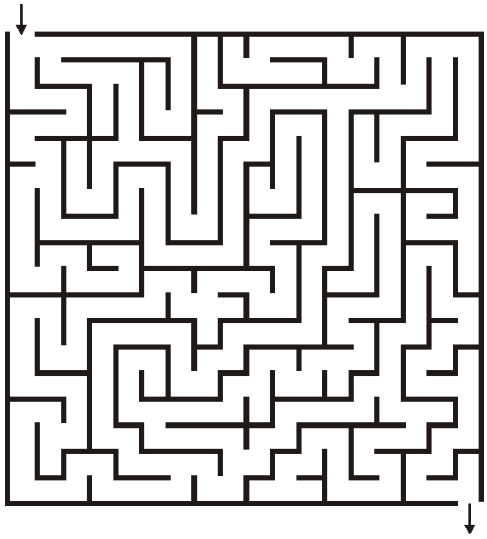

Figure 1: A simple maze.

(d) In our initial description of the problem we already abstracted from the real world, restricting actions and removing details. List three such simplifications we made.

**Question 3** (Question 4.1 of AIMA 2ed)
Using Figure 2, show the steps in expanding nodes for an A* search for the shorted path from Lugoj to Bucharest (using the straight line distance heuristic).

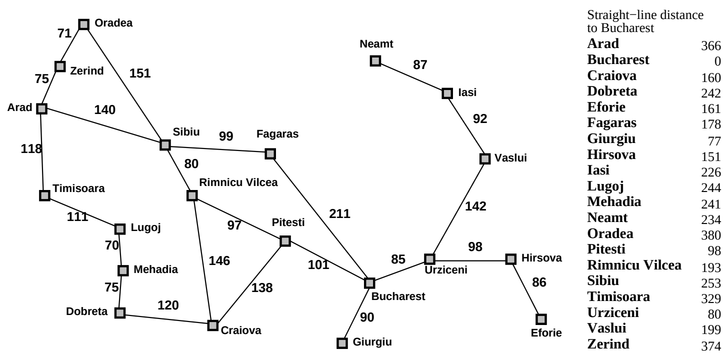


Figure 2: Map of Romania with straight-line distances.

**Question 4** (Question 4.2 of AIMA 2ed)
A heurisitic path algorithm is a best-first search algorithm with the objective function $f(n) = (2-w)g(n) + wh(n)$. Assuming $h(n)$ is admissible, then:

(a) For what values of $w$ is it guaranteed to be optimal?

(b) What kind of search is performed in the special case $w = 0$?

(c) What kind of search is performed in the special case $w = 1$?

(d) What kind of search is performed in the special case $w = 2$?

**Question 5** (Question 4.3 of AIMA 2ed)
Prove the following:

(a) Breadth first is a special case of uniform cost search.

(b) Breadth first, depth-first and uniform-cost searches are special cases of best-first search.

(c) Uniform-cost search is a special case of A* search.

**Question 6** (Question 4.5 of AIMA 2ed)
A "greedy" bestfirst approach to going from Iasi to Fagaras is at best suboptimal (it goes first to the deadend Neamt) and possibly fatally flawed (never gets to Fagaras) if repeated states are not detected (where it may oscillate between Vaslui and Neamt). Firstly, you should work this out for yourselves – assume the map in Figure 2 is reasonably indicative of the missing straightline distances (i.e., closeness on the map is actually closeness in real life). Likewise, note that there is no problem in solving the reverse trip – from Fagaras to Iasi. Then, answer the following question: are there problems for which the straightline distance heuristic fails in both directions (still in the context of best-first search)?