Artificial Intelligence
Wintersemester 17/18
Prof. Björn Ommer
ommer@uni-heidelberg.de

Exercise sheet 02: Uninformed and Informed search
Due on *10/11/2017*, 2pm.

Question 1: State Space I: Towers of Hanoi (2P)
Consider a special case of the game "Towers of Hanoi". It consists of three discs with different diameters and three rods. We denote the rodes with A, B, C and the discs with 1, 2, 3 in ascending order of diameters. We can move a disc from one rod onto another if the disc:

- is on top of the current rod
- after the moving the discs on the goal rod will be placed in ascending order of their diameters

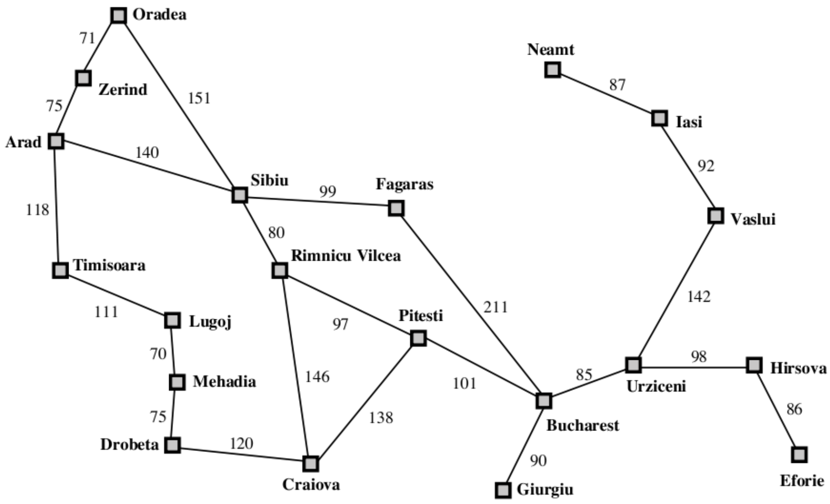
The goal is to move all three discs to the rod C.
Define the state-space representation of this problem (i.e, initial state, set of states, set of actions, goal test and path cost) and draw the corresponding state-space graph. When defining the actions formalize the conditions on a possible move.

Question 2: State Space II: Vacuum cleaner (1P)
Draw the complete state space graph with all possible paths for the following problem: the vacuum cleaner problem from the lecture. Now its sensor is broken and there is no information about its location. But the goal is still that there is no dirt on both places.
Hint: The starting state is the set of all 8 possible states since there is no location information.
Specify:

- maximum branching factor of the created graph
- depth of the least-cost solution (we assume that each performed action has a cost of 1)
- maximum depth of the state space

Question 3: Uninformed Search Algorithms I (1P)
Consider the state space graph from the previous question. Suppose the goal state is (left, clean, clean). List the content of the fringe in each step for

Prof. Björn Ommer
ommer@uni-heidelberg.de



Straight-line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Figure 1: Map of Romanian

- Breadth-first search (BFG)
- Depth-first search (DFS)
- Depth limited search with limit 3 (the root node has depth 0)

Question 4: Uninformed Search Algorithms II **(4P)**

Find the shortest way from Timisoara to Bucharest (see Fig. 1) using Uniform Cost Search. Solve the problem computationally by implementing the algorithm in Python. List the order in which nodes will be visited with the corresponding costs.

Note: additionally to the submitted code please include the listing of the search algorithm into the pdf-report.

please turn over

Question 5: Informed search algorithms: greedy best-first search (2P)
Consider the problem of getting to Bucharest. Assume than we posses additional information namely we know the straight-line distances (SLD) from each city to Bucharest. With this additional information we can think about a search algorithm that expands the most desirable (i.e the closest to the goal) node. Such algorithm is called *greedy best-first search.*
Example: We start again in Arad (SLD=366). From Arad we can go further to Zerind (SLD=374), Sibui (SLD=253) or Timisoara (SLD=329). The greedy best-first search algorithm will select Sibui as the next aim, as it is closer to Bucharest than the other two cities.
Answer two following questions about the greedy best-first search:

a) Is there a city from which there is no connection to Bucharest using a greedy best-first search?
Hint: You don't need to apply the algorithm on all starting cities! A lot of cities can be excluded looking at the graph. Write down the reasoning for your solution!

b) Provide the names of the start cities for that greedy best-first search does not find an optimal path (here: shortest path) to Bucharest.
Hint: Again a lot of cities can be excluded without actually applying the algorithm on paper. Write down the reasoning for your solution!

Note: Submit exactly one ZIP file and one PDF file via Moodle before the deadline. The ZIP file should contain your executable code. Make sure that it runs on different operating systems and use relative paths. Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. The PDF file should contain your written code, all figures, explanations and answers to questions. Make sure that plots have informative axis labels, legends and captions.