Introduction to Artificial intelligence

Tutorial Sheet 3 (Searching)

Exercise 1

Give a complete problem formulation for each of the following. Choose a formulation that is precise enough to be implemented.

a. Using only four colors, you have to color a planar map in such a way that no two adjacent regions have the same color.

b. A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates.

c. You have a program that outputs the message “illegal input record” when fed a certain file of input records. You know that processing of each record is independent of the other records. You want to discover what record is illegal.

d. You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly one gallon.

Exercise 2

Consider a state space where the start state is number 1 and each state k has two successors: numbers 2k and 2k + 1.

a. Draw the portion of the state space for states 1 to 15.

b. Suppose the goal state is 11. List the order in which nodes will be visited for breadth first search, depth-limited search with limit 3, and iterative deepening search.

c. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?

d. Does the answer to (c) suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?

e. Call the action going from k to 2k Left, and the action going to 2k + 1 Right. Can you find an algorithm that outputs the solution to this problem without any search at all?

Exercise 3

A basic wooden railway set contains the pieces shown in Figure 3.32. The task is to connect these pieces into a railway that has no overlapping tracks and no loose ends where a train could run off onto the floor.

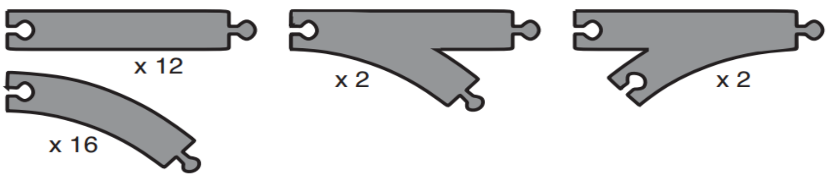

Figure 3.32 The track pieces in a wooden railway set; each is labeled with the number of copies in the set. Note that curved pieces and “fork” pieces (“switches” or “points”) can be flipped over so they can curve in either direction. Each curve subtends 45 degrees.

a. Suppose that the pieces fit together *exactly* with no slack. Give a precise formulation of the task as a search problem.

b. Identify a suitable uninformed search algorithm for this task and explain your choice.

c. Explain why removing any one of the “fork” pieces makes the problem unsolvable.

d. Give an upper bound on the total size of the state space defined by your formulation. (*Hint*: think about the maximum branching factor for the construction process and the maximum depth, ignoring the problem of overlapping pieces and loose ends. Begin by pretending that every piece is unique.)

Exercise 4

Describe a state space in which iterative deepening search performs much worse than depth-first search (for example, O(n^2) vs. O(n)).

Exercise 5

Consider the unbounded version of the regular 2D grid shown in Figure 3.9. The start state is at the origin, (0,0), and the goal state is at (x, y).

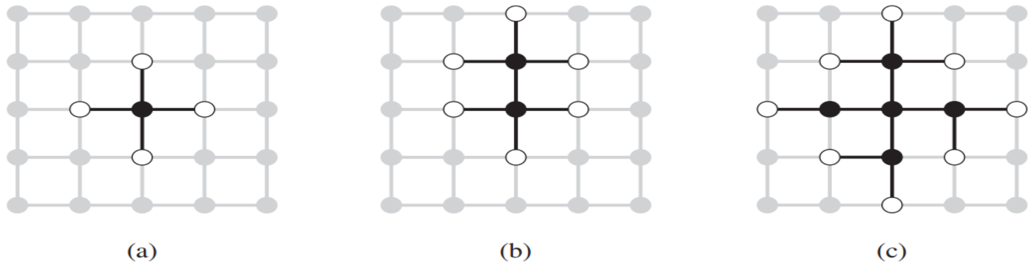

Figure 3.9 The separation property of GRAPH-SEARCH, illustrated on a rectangular-grid problem. The frontier (white nodes) always separates the explored region of the state space (black nodes) from the unexplored region (gray nodes). In (a), just the root has been expanded. In (b), one leaf node has been expanded. In (c), the remaining successors of the root have been expanded in clockwise order.

a. What is the branching factor b in this state space?
b. How many distinct states are there at depth k (for $k > 0$)?
c. What is the maximum number of nodes expanded by breadth-first tree search?
d. What is the maximum number of nodes expanded by breadth-first graph search?
e. Is $h = |u - x| + |v - y|$ an admissible heuristic for a state at (u, v)? Explain.
f. How many nodes are expanded by A∗ graph search using h?
g. Does h remain admissible if some links are removed?
h. Does h remain admissible if some links are added between nonadjacent states?

Exercise 6

Suppose two friends live in different cities on a map, such as the Romania map shown in Figure 3.2.

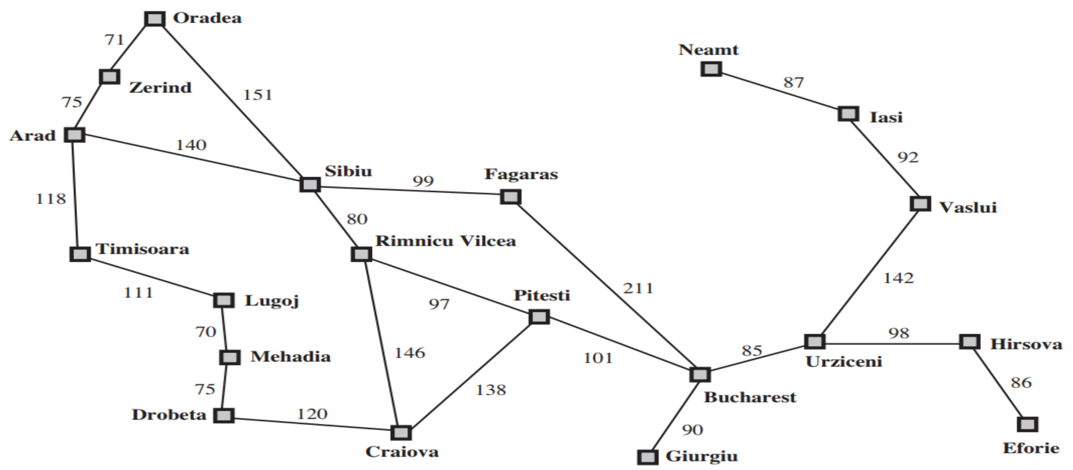

Figure 3.2 A simplified road map of part of Romania.

On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the road distance d(i, j) between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

a. Write a detailed formulation for this search problem. (You will find it helpful to define some formal notation here.)
b. Let D(i, j) be the straight-line distance between cities i and j. Which of the following heuristic functions are admissible? (i) D(i, j); (ii) 2 * D(i, j); (iii) D(i, j)/2.
c. Are there completely connected maps for which no solution exists?
d. Are there maps in which all solutions require one friend to visit the same city twice?

Exercise 7

Which of the following are true and which are false? Explain your answers.

a. Depth-first search always expands at least as many nodes as A∗ search with an admissible heuristic.

b. h(n) = 0 is an admissible heuristic for the 8-puzzle.

c. A∗ is of no use in robotics because percepts, states, and actions are continuous.

d. Breadth-first search is complete even if zero step costs are allowed.

e. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

Exercise 8

Trace the operation of A∗ search applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the f, g, and h score for each node.

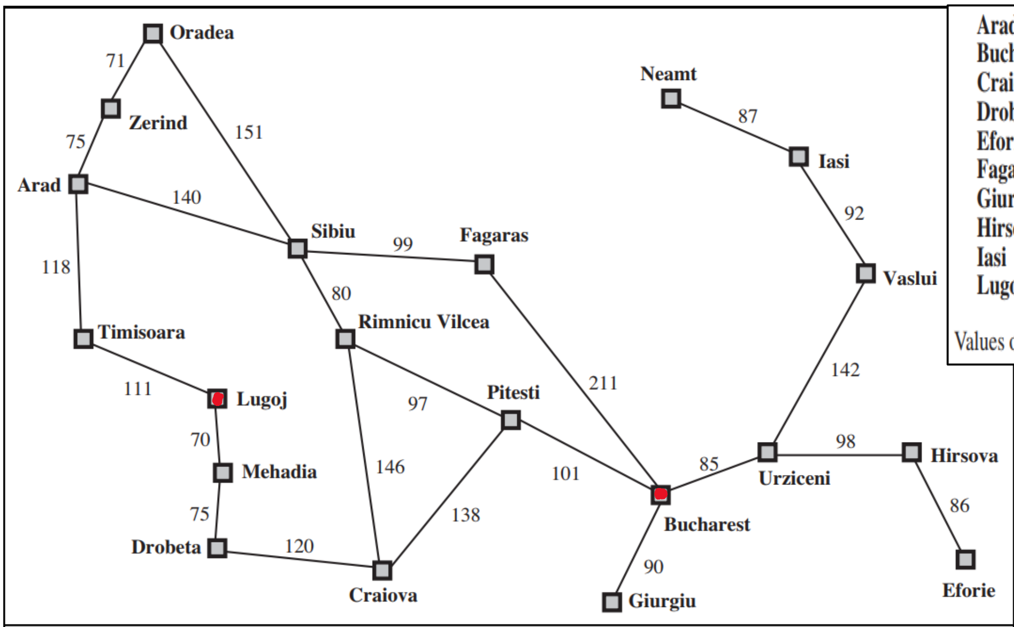

Figure 3.2 A simplified road map of part of Romania.

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

Values of h_{SLD}—straight-line distances to Bucharest.