Problem-solving by search exercises

(Source: https://aimacode.github.io/aima-exercises/, accessed in Oct 2023)

Exercise 01

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

1. Depth-first search always expands at least as many nodes as A search with an admissible heuristic.
2. h(n)=0 is an admissible heuristic for the 8-puzzle.
3. A is of no use in robotics because percepts, states, and actions are continuous.
4. Breadth-first search is complete even if zero step costs are allowed.
5. 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.

Solution (textbook manual)

1. **False**: a lucky DFS might expand exactly d nodes to reach the goal. A∗ largely dominates any graph-search algorithm that is guaranteed to find optimal solutions.
2. **True**: h(n) = 0 is always an admissible heuristic, since costs are nonnegative.
3. **True**: A* search is often used in robotics; the space can be discretized or skeletonized.
4. **True**: depth of the solution matters for breadth-first search, not cost.
5. **False**: a rook can move across the board in move one, although the Manhattan distance from start to finish is 8.

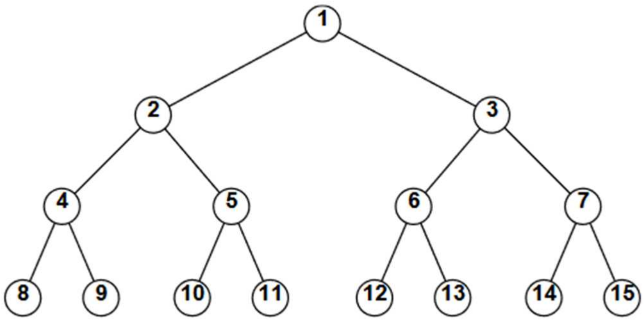
Exercise 02

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

1. Draw the portion of the state space for states 1 to 15.
2. 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.
3. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?
4. 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?
5. 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?

Solution (textbook manual)

1. See Figure



2. Breadth-first: 1 2 3 4 5 6 7 8 9 10 11
   Depth-limited: 1 2 4 8 9 5 10 11
   Iterative deepening: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11
3. Bidirectional search is very useful, because the only successor of n in the reverse direction is |(n/2)|. This helps focus the search. The branching factor is 2 in the forward direction; 1 in the reverse direction.
4. Yes; start at the goal, and apply the single reverse successor action until you reach 1.
5. The solution can be read off the binary numeral for the goal number. Write the goal number in binary. Since we can only reach positive integers, this binary expansion beings with a 1. From most- to least- significant bit, skipping the initial 1, go Left to the node 2n if this bit is 0 and go Right to node 2n + 1 if it is 1. For example, suppose the goal is 11, which is 1011 in binary. The solution is therefore Left, Right, Right.

Exercise 03

For each of the following assertions, say whether it is true or false and support your answer with examples or counterexamples where appropriate:

1. A hill-climbing algorithm that never visits states with lower value (or higher cost) is guaranteed to find the optimal solution if given enough time to find a solution.
2. For any local-search problem, hill-climbing will return a global optimum if the algorithm is run starting at any state that is a neighbor of a neighbor of a globally optimal state.

Solution (textbook manual)

1. False. Such an algorithm will reach a local optimum and stop (or wander on a plateau).
2. False. The intervening neighbor could be a local minimum and the current state is on another slope leading to a local maximum. Consider, for example, starting at the state valued 5 in the linear sequence 7,6,5,0,9,0,0.