Problem 1

3.14 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 precepts, 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.

a. False. Depth-first search may by chance find the goal state without needing to travel back. In which case it expand less nodes than A^* search.

b. True. Admissible heuristics never overestimate thus $h(n) = 0$ is admissible for the 8-puzzle.

c. False. While precepts, states, and actions are continuous they can be made discrete.

d. True. Cost has no effect on the solution.

e. False. Manhattan distance measures the number of squares as distance. For example, four squares. However, a rook can move anywhere from one to eight squares in a single move.

Problem 2

3.15 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.

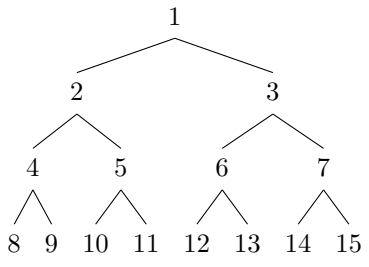
b. Suppose the goal state is 11. List the order in which nodes will be visited for breadth-first search, depth-limited search with limit 3, and iterative deepening search.

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 what 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?

a.



b. Breadth-first: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
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

c. Very well, the branching factor in the reverse direction (1) is less than the forward direction (2).

d. Reversing the problem, making 11 the start state and 1 the goal state will result in barely any search site in this direction the branching factor is 1.

e. Converting the goal state to binary (e.g., 11 to 1011). Starting from left to right, skipping the left most digit we go left if we find a 0 and right if 1.

Problem 3

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

Imagine a tree than looks like the following:

A
|
B
|
C
|
D
|
E

Performing a Depth-first search results in the following:
A, B, C, D, E

Performing a Iterative deepening search results in the following:
A
A, B
A, B, C
A, B, C, D
A, B, C, D, E

In this case Iterative deepening performs much worse than Depth-first search.

Problem 4

3.21 Prove each of the following statements, or give a counterexample:

a. Breadth-first search is a special case of uniform-cost search.

b. Depth-first search is a special case of best-first tree search.

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

a. If we make all the edge weights equal then uniform-cost will be identical to breadth-first.

b. If we perform a best-first search with a heuristic evaluation function $f(n)$ that prefers nodes with the greatest depth, then it will behave like a depth-first search.

c. Uniform-cost search is a A^* search without a heuristic, essentially $h(n) = 0$ always.

Problem 5

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