Michael Chillemi
Homework 2

Topics Covered: This assignment covers: Uninformed Search.
Must be done independently.

Complete the following problems/answer the following questions.

Problem 1:

A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. She would like to get the bananas and specifically end up on the ground with the bananas. The room contains two stack-able, movable, climbable 3-foot-high crates, which you can call C1 and C2. The monkey is initially on the ground, as are both of the crates, and nothing is under the bananas initially. Assume that the monkey wants to accomplish their task with the fewest possible actions. Give a complete problem formulation as a search problem, precise enough to be implemented. Recall that our definition of a search problem involves specifying the set of all states, a non-empty set of start states, a non-empty set of goal states, the successors function that maps each state onto the set of states that are reachable in a single action, and a cost function mapping state transitions to their costs, which for this problem would just have a cost of 1 for each transition. In answering this question, do the following:

a. Develop notation for describing a state precisely (i.e., indicating where everything is, whether the monkey is holding the bananas, etc). Try to find a concise notation for this.

Answer = {Banana Status, Monkey Status,C1 Status, C2 Status}

{B = has bananas N = No bananas, G = on the ground C1 = on top of C1 C2 = on top of C2, C2 = on top of C2 G = on the ground with nothing on top of it U = on the ground under a box, C1 = on top of C1 G = on the ground with nothing on top of it U = on the ground under a box}

b. In your notation, specify the start state.

Answer =

START STATE = {N,G,G,G}

c. In your notation, specify the goal state.

Answer =

GOAL STATE = {B,G,U,C1}
{B,G,C2,U}
{B,G,G,G}

d. Draw a graph that shows ALL states in the problem and all transitions. That is, the graph should have a node for each state from the set of all states, and use directed edges to show the successors function (i.e., you'll have a directed edge from one node to another to show that a state is a successor of another).

Problem 2: Recall that the average branching factor for a search problem is the average number of successors of a state (averaged over all problem states).

a. Compute the average branching factor for the 8-puzzle (sliding tile puzzle with a 3 by 3 grid). Show your work (don't just indicate the branching factor).

b. Compute the average branching factor for the 15-puzzle (sliding tile puzzle with a 4 by 4 grid). Show your work (don't just indicate the branching factor).

c. What is the branching factor of a Rubik's Cube (assuming that an action consists of rotating any side a quarter turn in any direction)? Explain your answer.

Problem 3: Consider a state space where the start state is number 1 and each state k has two successors: numbers 2k and 2k+1. As you answer the questions that follow, assume that this state space is infinite in size.

a. Draw the portion of the state space for states 1 to 15. Put the 2k successor as the left child of state k and the 2k+1 child as the right child of state k.

b. Suppose the goal state is 11. List the order in which nodes will be visited by breadth-first search, assuming that the successors are given in increasing order (i.e., you visit the left 2k successor before the 2k+1 successor).

c. Suppose the goal state is 11. List the order in which nodes will be visited by depth-limited search with limit 3, assuming that the successors are given in increasing order(i.e., you visit the left 2k successor before the 2k+1 successor).

d. Suppose the goal state is 11. List the order in which nodes will be visited by iterative deepening search, assuming that the successors are given in increasing order (i.e., you visit the left 2k successor before the 2k+1 successor).

e. Again, suppose that the goal state is 11. Will depth-first search terminate, assuming that the 2k successor is always visited before the 2k+1 successor?

f. Again, suppose that the goal state is 11. Will depth-first search terminate, assuming that the 2k+1 successor is always visited before the 2k successor?

g. What is the branching factor for this search space?

h. We were given the successors function. It is actually straightforward to specify a predecessors function as well. What are the predecessor(s) of state k?

i. Now that we have a predecessors function, what is the branching factor in the backwards direction for this search space (i.e., if we were backward chaining from the goal)?

j. Describe the search algorithm that will solve this problem with the least amount of search of all possible search algorithms. Explain your answer. If your answer is an algorithm we've seen, you can just name it, and provide an explanation for why no other search algorithm can solve this problem with less search. If it is a variation of a search algorithm we've seen, you can just indicate its features, and provide an explanation for why no other search algorithm can solve this problem with less search. Hint: Think carefully about your answers to previous parts of this problem.