CS:4420 Artificial Intelligence
Spring 2019

Homework 2

Part B

Due: Friday, Feb 22 by 11:59pm

This assignment has two parts, A and B, both to be done *individually.* This document describes Part B which consists of written assignments. Write your answers in a text editor or word processor as you prefer, but submit them as a file in PDF format on ICON. Make sure to clearly mark your question with its problem number an to *write your name in the file.*

Pay particular attention to these points:

- *Pay close attention to the specification of each problem and the restrictions imposed on its solution.* Solutions ignoring the restrictions may receive only partial credit or no credit at all.

1 Search strategies

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

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. In chess, Manhattan distance is an admissible heuristic for the problem of moving a rook from square A to square B in the smallest number of moves.[1]
6. Suppose that for a given problem you have an admissible heuristic h. Let h' be such that $h'(n) = h(n) + k$ for every node n where k is a positive constant. Then, we are guaranteed to find the optimal solution even if we use A* with h' instead of h.

1.1 Practice with search strategies

Consider the search tree given in Figure 1. The letter inside a node n is the name of the (state represented by that) node. The subscript of the letter is the heuristic estimate $h(n)$ of the cost of

[1] Recall 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.

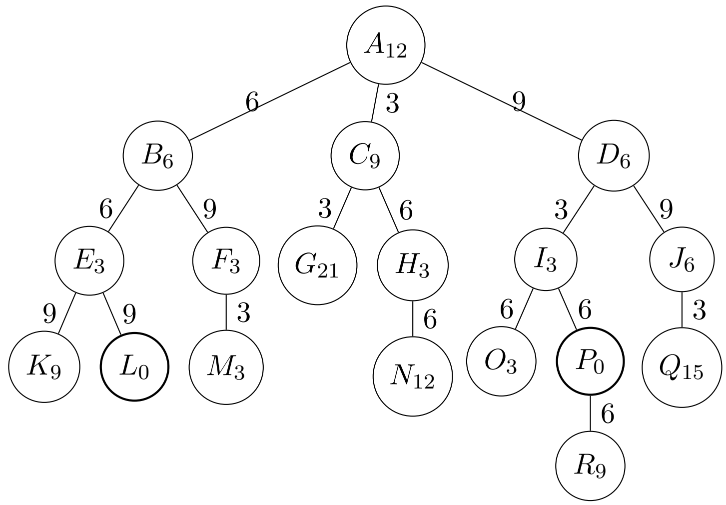

Figure 1: Search tree for Problem 1

getting from n to the least-cost goal. The number on each edge (n_1, n_2) is the step cost of going from the state in n_1 to the state in n_2. The actual cost of going from a node n_1 to a node n_2 along a connecting path p is the sum of the step costs along p. The initial node is the root node A. Goal nodes are represented by thicker circles.

1. Using the general tree-search algorithm in Figure 3.7 of the textbook, perform a search of the tree according to each of the following search strategies. Make sure to *follow the algorithm, not just your intuitive understanding of each search strategy.*

   (a) Breadth-first
   (b) Uniform cost
   (c) Greedy best-first
   (d) A*
   (e) Hill-climbing (For this tree, smaller values are better)
   (f) Local beam search with number of nodes $k = 2$ and *initial nodes B and D* (Again, smaller values are better).

   Treat the frontier as a priority queue ordered by an appropriate evaluation function.

   For each strategy:

   - Specify what evaluation function and additional restrictions on the queue are needed for the general algorithm in Figure 3.7 to implement that strategy.
   - Show the nodes in the order they are removed from the queue (the frontier) for expansion. For each removed node, show the queue produced after the expansion of the node. Write the queue from left to right. When you order the queue, if two nodes have the same value, sort them alphabetically. Add to each node, as a subscript, the value being used to sort that node, if any.

Your solution should look something like:

Node expanded	Queue
-	(A_0)
A_0	$(B_5\ C_6\ D_7)$
B_5	$(C_6\ D_7\ E_8)$
...	...

showing the node expanded at a particular time step, and the queue resulting from expanding that node.

2. Which strategies found the optimal (least-cost) solution to the problem?