Artificial Intelligence Spring 2019 Homework 1: Search

Please follow this format for submission. You may reproduce it in LaTeX or word if you wish

Name: Vikram.

UNI: HU.

WRITTEN

Please justify all your answers to receive full credit (unless stated otherwise). There are 5 questions.

Question 1: PEAS

A 4-tower Hanoi puzzle consists of three pegs and 4 disks of different sizes which can slide onto any peg. The objective of the puzzle is to move the entire stack to the right peg. You can only move one disk at a time. Each move consists of taking the upper disk from one of the pegs and placing it on top of another stack, and no disk may be placed on top of a smaller disk.

Initial State

Goal State

1. What elements are necessary to formally define a specific search problem?

Problem formula : The elements are

1) Initial state
2) A set of states
3) A set of Actions
3) A transition model
4) goal test
5) path cost.

2. Formulate the search problem for the 4-tower of Hanoi puzzle shown above. Be specific.

1) Initial state : All the disks is on the 1st peg (leftmost peg)
2) A set of states : all possible states
3) A set of Actions: moving the disk from one peg to another
4) Transition model : Given a state and an action, returns the resulting state

3. What is the size of the state space?

The size of the state space with 4 disks is $3^4 = 81$

Since there are 3 possible ways to move each disk

For n number of disk, it is 3^n

5) Goal Test : Are all 4 disk organized from smallest at

top to largest at bottom in any of the pegs that are not the peg at the start,

6) Performance cost - All moves it took to reach the goal, each move is 1 cost.

Question 2

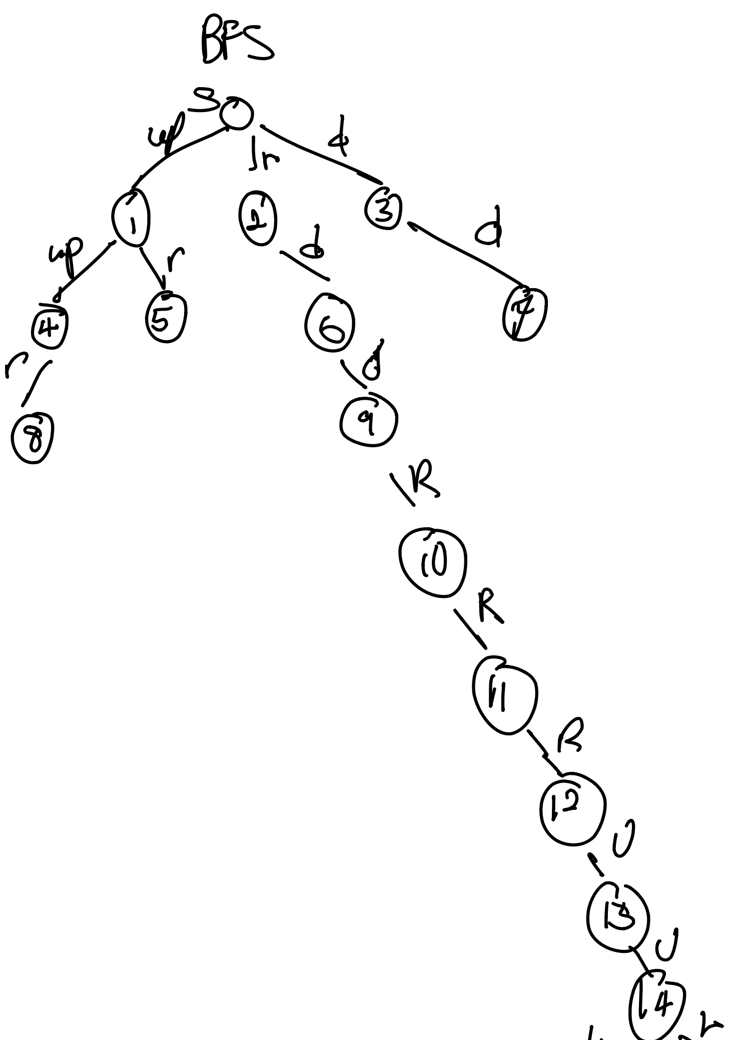
1. Given a maze with the start state S and the goal state G. Number the cells (starting from 1) in the order they are visited by BFS and DFS using the following guidelines: Gray cells contain obstacles and cannot be used. Moves are considered in the following order: Up, Right, Down, Left. Diagonal moves are not allowed. No state is visited twice. No need to justify.

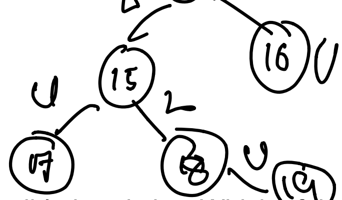
a. Breadth-First search

4	8		G 19	17
1	5		18	15
S	2		16	14
3	6			13
7	9	10	11	12

b. Depth-First search

2	3		G 15	14
1	4			13
S	5			12
	6			11
	7	8	9	10





Question 3

Let h1 and h2 be two admissible heuristics. Which of the following heuristics are admissible?

(a) $h(n) = min\{5 * h1(n), h2(n)\}$

Since $h1(n)$ and $h2(n)$ is admissible

$h1(n) \leq h^*(n)$, $h2(n) \leq h^*(n)$

This implies

$5 * h1 > h^*(n)$ therefore

$h(n)\ min\{5 * h1(n), h2(n)\} = h(2)$ which is admissible

(b) $h(n) = max\{h1(n), h2(n)\}$

Since $h1(n)$ and $h2(n)$ is admissible

$h1(n) \leq h^*(n)$, $h2(n) \leq h^*(n)$

$max\{h1(n), h2(n)\} \leq h^*(n)$

so $h(n)$ is admissible

(c) $h(n) = w * h1(n) + (1 - w) * h2(n)$, with $0 \leq w \leq 1$

Since $h1(n)$ is admissible   since $h2(n)$ is admissible

$h1(n) \leq h^*(n)$ $h2(n) \leq h^*(n)$

$0 \leq w * h1(n) \leq w * h^*(n)$

$0 \leq (1-w) * h2(n) \leq (1-w) * h^*(n)$

$h(n) \leq w * h^*(n) + (1-w) * h^*(n)$

$h(n) \leq h^*(n) \rightarrow h(n)$ is admissible

(d) $h(n) = \frac{max\{2 * h1(n), h2(n)\}}{2}$

Since $h1$ is admissible   Since $h2$ is admissible

$h1(n) \leq h^*(n)$ $h2(n) \leq h^*(n)$ — (1)

$2 * h1(n) > h^*(n)$ (2)

From (1) and (2)

$max\{2 * h1(n), h2(n)\} = 2 * h1(n)$

$h^*(n) < 2 * h1(n) \leq 2 * h1(n)$

$\frac{h^*(n)}{2} < \frac{2 * h1(n)}{2} \leq h1(n) \leq h^*(n)$

so $h(n) = \frac{max\{2 * h1(n), h2(n)\}}{2}$ is admissible

Question 4

Provide three advantages and three disadvantages of using local search algorithms. Explain.

Advantages

1. No need to maintain a search tree
Local search algorithms operate using a single current node. and generally move only to neighbours of that node.

2. Use very little memory
Do not have to maintain a search tree and usually need not keep track of paths.

3. can often find good enough solutions or large state spaces

Disadvantages

1. Lack of domain knowledge
- stuck in a local maxima if goal was global maxima, e.g in the hill climbing search
- when the state-space landscape has local minima, any search that moves only in greedy direction cannot be complete.

2. Possibility of infinite loops

Question 5

The 8-puzzle is a 3×3 board with 8 tiles (corresponding to 3^2 - 1), numbered from 1 to 8, and one empty space. The 8-puzzle is a special case of the N-puzzle containing N = m^2 - 1 tiles and one empty space. Answer the following questions. Justifications may include for example, drawing a partial search tree, providing counter examples, etc. <u>A rigorous justification will receive extra credit.</u> A good starting point is: "Notes on the 15 Puzzle," American Journal of Mathematics, Vol. 2, No. 4 (Dec., 1879), pp. 397-404. Feel free to research the topic in other sources, but cite your references.

1. Are all instances of the 8-puzzle games solvable?

Not all instances of 8-puzzle is solvable. To answer this, we look at the parity of inversion. consider 123456 . 123456 has 0 inversion.

1	2	3
4	5	6
0	8	7

Ignoring the blank space, 0

but 87 has one inversion. as 8 is a bigger number preceding 7. since total number of inversion is 1 it is not solvable.

2. Provide an upper bound on the size of the state space in the N-puzzle game? Consider only reachable configurations. Note: A tight bound will receive extra credit.

The size of the state space in the N puzzle-game is n!.

9	8	7
6	5	4
3	2	1

where each tile shows the number of possible entries of each tile given no repetition is allowed. so we end up with 9!.

n! is the sum of reachable and non reachable configurations since to be reachable, every

tile needs to satisfy the parity of inversions
ie. every tile needs another tile before it to swap
with. We know only even number of permutations
are solvable. so the bound of the state
space is $\frac{n!}{2}$

$\vdots$