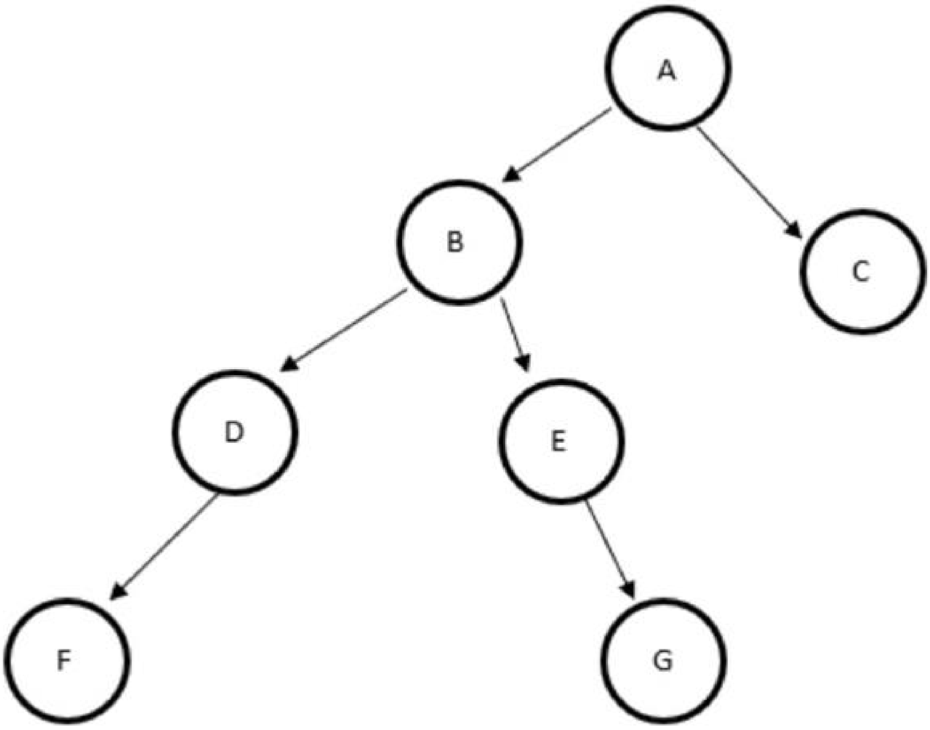
Mid-Semester Test

Question 1

1 / 1 pts

Consider the search tree shown in the figure below. Assume the Goal state is G and that ties are broken alphabetically (e.g. B before C). Using Iterative Deepening Search on the tree above, how many times will node B be visited before a solution is found?

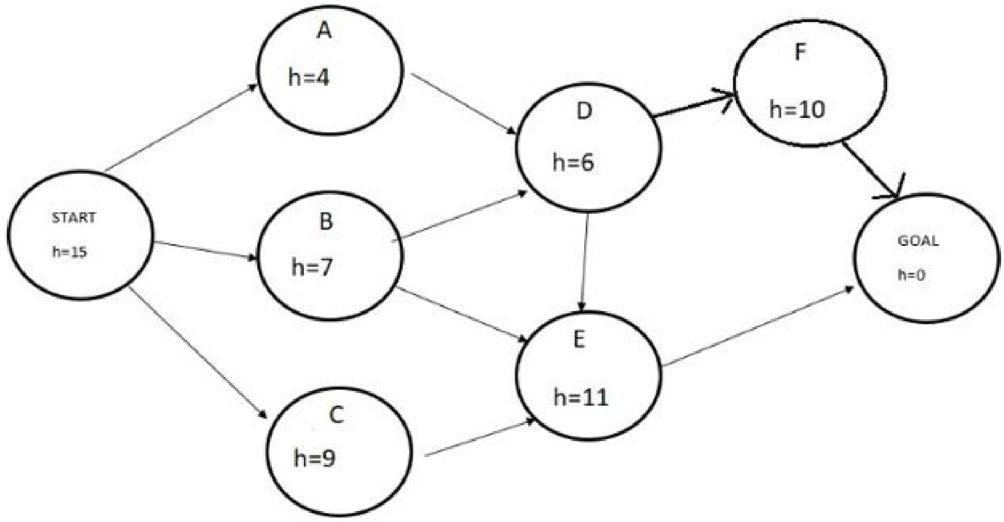


3 times

Question 2

0 / 1 pts

Consider the search tree shown in the diagram below. The first node expanded will be the START node. Using the Weighted A* Algorithm with *W=3* and assuming a uniform cost of 2 to move between nodes, which will be the 5th node expanded?



C (The first node expanded is the START node)

Question 3

0 / 1 pts

Assume a robot is situated in a 8 x 8 grid. The robot can move left, right, up and down within the grid. Each of these actions has a cost of 1. The robot keeps track of its current position using two fluents row(X) and col(X). For example, if the robot was in position (3, 4) in the grid then the fluents row(3) and col(4) would be true.

Assume the robot is initially in the state {row(1), col(1)} and that the goal is to reach {row(6), col(6)}. What is the h^{max} value for the initial state?

Max(5, 5) = 5

Question 4

1 / 1 pts

Below is the Bellman-Ford Table for $h^{add}(I)$ for a particular problem where I is the initial state of the problem.

A	B	C	D
Infinity	Infinity	Infinity	Infinity
6	8	Infinity	8
6	5	3	4
6	5	3	4

If the goal is {A, B}, what is the value of $h^{add}(I)$?

6 + 5 = 11

Q5: The h+ heuristic can be calculated by:

- Solving the delete relaxed problem using a satisficing planner
- Taking the average of the hadd and hmax heuristics
- Solving the original problem using a satisficing planner
- Solving the delete relaxed problem using an optimal planner ✓
- Counting the number of goal atoms are not true in the current state

Q6: Consider a heuristic function that sets the heuristic value for each state to zero (i.e. the behavior of the null Heuristic from your first assignment). For all search problems with positive action costs, this heuristic is:

- Admissible, Consistent and Goal Aware, but not necessarily safe
- Admissible, Consistent and Safe, but not necessarily Goal Aware
- Admissible, Goal Aware and Safe, but not necessarily Consistent
- Consistent, Goal Aware and Safe, but not necessarily Admissible
- Admissible, Goal Aware, Safe and Goal Aware ✓
- None of the other answers

Q7: Which of the following statements are true (select all that apply)?

- All safe heuristics are goal aware
- The h+ heuristic is admissible for all search problems ✓
- Breadth first search is complete for all search spaces ✓
- The goal counting heuristic is admissible for all search problems
- Iterative Deepening Search is complete for all search spaces ✓
- The hadd heuristic is admissible for all search problems

Q8: Which of the following statements are true (select all that apply)?

- Satisficing plans are generally more difficult to compute than optimal plans
- Both PlanLen and PlanEx are PSPACE-complete in general ✓
- The relaxation produced by removing preconditions and delete lists form a STRIPS planning problem is efficiently computable
- The relaxation produced by removing delete lists from a STRIPS planning problem is efficiently constructable ✓
- Any STRIPS planning problem can be modelled using PDDL ✓

Q9: Consider the Pacman domain used in your first assignment, where the goal is to eat all of the food on the map. Imagine that eating a food dot caused Pacman to move one of the unoccupied adjacent nodes (e.g. the one above, below, left or right of the current position), with an equal probability of moving to any of those nodes. This could best be modelled as:

- A Markov Decision Process ✓
- A Partially Observable Markov Decision Process
- A Boolean Satisfiability Problem
- A Classical planning problem

Q10: Consider the well-known blocks world domain used in lectures. When modelled using STRIPS, how many predicates will appear in the delete list for the *pickup* action?

3 (clear(x), onTable(x), armFree)

Functions of the blocks world (From Tutorial 4)

$$F := \{on(x,y), onTable(x), clear(x), holding(x), armFree\}$$
$$O :=$$
$$\{stack(x,y) :=$$
$$-prec := \{holding(x), clear(y)\}$$
$$-add := \{clear(x), on(x,y), armFree\}$$
$$-del := \{clear(y), holding(x)\}$$
$$|x, y \in B\}$$
$$\cup \{$$
$$unstack(x,y) :=$$
$$-prec := \{on(x,y), clear(x), armFree\}$$
$$-add := \{holding(x), clear(y)\}$$
$$-del := \{clear(x), on(x,y), armFree\}$$
$$|x, y \in B\}$$
$$\}$$
$$\cup \{$$
$$putdown(x) :=$$
$$-prec := \{holding(x)\}$$
$$-add := \{clear(x), onTable(x), armFree\}$$
$$-del := \{holding(x)\}$$
$$|x \in B\}$$
$$\}$$
$$\cup \{$$
$$pickup(x) :=$$
$$-prec := \{onTable(x), clear(x), armFree\}$$
$$-add := \{holding(x)\}$$
$$-del := \{clear(x), onTable(x), armFree\}$$
$$|x \in B\}$$
$$\}$$
$$I := \{on(A,C), onTable(C), onTable(B), clear(A), clear(B), armFree\}$$
$$G := \{on(A,B), on(B,C)\}$$

where, B is the set of blocks