CSC384 Test 1 Sample Questions

October 11, 2018

1 Short Answer

1. Is A*'s search behavior **necessarily** "exponentially explosive"?. That is, does its search time **always** grow at least exponentially with the length of the optimal solution.

2. It would seem that iterative deepening search should have a higher asymptotic time complexity than breadth-first search because every time the depth-bound is increased it must start its search from scratch. However this is not true. Why?

3. If $h()$ is admissible and s is the start node, how is $h(s)$ related to the cost of the solution found by A* search?

4. Let $h^*(n)$ denote the cost of an *optimal* path from n to a goal state (i.e., an optimal path from the end state of n to a state satisfying the goal). It can be shown (see question below) that when $h(n) = h^*(n)$ A* only expands nodes that are prefixes of an optimal path to a goal. Can A* still require exponential time to solve a problem when $h(n) = h^*(n)$? What addition to A* do we need to ensure that A* takes only time linear in the length of the solution when $h(n) = h^*(n)$?

5. Consider solving the Sudoku problem using A* search. The start state has some number of cells filled in, and the successors of each state are all legal ways an additional cell can be filled in. At any time we know exactly the number of cells that remain to filled. Therefore, if K cells remain to be filled at node n, we know that $h^*(n) = K$. Thus if we set our heuristic function $h(n)$ to be $h^*(n) = K$, we will have perfect heuristic knowledge in A^*.

   Explain the mistake in the above statement.

6. What happens if we use a heuristic $h()$ in A* search that does not have the guarantee that $h(n) \leq h^*(n)$ for all paths n.

7. Is the following game given in normal form zero sum or not?

			Player II	
		I	II	III
	I	0/0	-1/1	1/-1
Player I	II	1/-2	0/1	-1/1
	III	-1/1	1/-1	0/0

2 Search Space Representation

1. Consider the problem of transporting N people across a river, where each person has a certain weight. Everyone starts off on the right-hand side of the river and is to be transported to the left-hand side. Suppose also that only one boat exists, which is able to support a maximum weight of W_B.

   (a) Design a state space representation for this problem. Specify clearly the meaning of each component of the state representation.

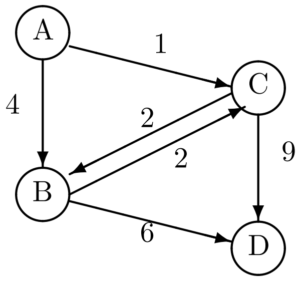
   (b) Design a complete set of operators in this domain based on your choosen state state representation. Note that these operators should also work for transporting any allowed subgroup of people in either direction. That is, they should not be overly specific to the actual problem we want to solve, but instead should allow one to traverse the search space in a general manner.

3 Search Algorithms

1. Trace the execution of A* for the graph shown below.

   - Show the successive configurations of the frontier where the elements on the frontier are paths. That is, the path $s_1 \rightarrow s_2 \rightarrow s_3$ would be written $[s_1, s_2, s_3]$. Under each node of the frontier, indicate the f, g and h values of each node (e.g., if $g(n) = 5$ and $h(n) = 7$ write "12=5+7" underneath n.
   - Indicate the path that is expanded at each stage.
   - Finally show the path to the goal found by A^*.

- A is the start state and D is the goal state.
- The table of h values is shown. For example, every path ending at state A has heuristic value of 8.



h(A) = 8
h(B) = 3
h(C) = 7
h(D) = 0

(a) Indicate how the search would change if cycle checking is used, and how it would change if path checking is used.

(b) Is the heuristic admissible? Is it monotonic?

4 Simple Proofs

1. Prove that if $h(n) = h^*(n)$ for all n, then whenever A^* expands a node n, n must lie on an optimal path to a goal.

2. Prove that when $h(n)$ is montone, then the f value of any path n must be at least as large as the f value of any prefix of n. (So the f value is monotonic non-decreasing as we extend a path). More specifically: for any path $n = [s_0, s_1, ..., s_k]$, and any prefix n_i of n with $n_i = [s_0, s_1, ..., s_i]$ $(i \leq k)$ we must have that $f(n) \geq f(n_i)$ when h is monotone.

5 Backtracking Search/CSPs

1. A latin square of size m is an $m \times m$ matrix containing the numbers 1–m such that no number occurs more than once in any row or column. For example

```
1 2 3 4
4 1 2 3
3 4 1 2
2 3 4 1
```

   is a latin square of size 4. Specify this the problem of finding a latin square of size m as a CSP.

2. Consider the N-Queens problem. That is, the problem of placing N Queens on an $N \times N$ chess-board so that no two Queens can attack each other.

   Find the *first solution* to the 5-Queens problem by using the *Forward Checking* algorithm using *Minimum Remaining Values* heuristic (always instantiate next the variable with smallest remaining number of elements in its current domain). Also breaking ties in favour of the lowest numbered variable.

(Note, use the same CSP formulation as that used in class. That is, we have 5 variables, $Q_1, \dots, Q_5$ each with domain $[1, 2, 3, 4, 5]$. Each variable Q_i represents the queen in the i-th row, and the assignment $Q_i = j$ means that the queen in the i-th row has been placed in the j-th column.

Draw the search tree explored by this algorithm. **At each node** indicate

(a) The variable being instantiated, and the value it is being assigned.

(b) A list of the variables that have had at least one of their values pruned by the new assignment, and for such variable a list of its remaining legal values. *Note, you must follow the forward checking algorithm precisely: only prune values that would be pruned by the algorithm.*

(c) Mark any node where a deadend occurs because of a domain wipe out (use the symbol DWO).

3. Say we have 4 variables X, Y, Z, and W, with the following domains of values:

   (a) $Dom[X] = \{1, 2, 3, 4\}$

   (b) $Dom[Y] = \{1, 2, 3, 4\}$

   (c) $Dom[Z] = \{1, 2, 3, 4\}$

   (d) $Dom[W] = \{1, 2, 3, 4, 5\}$

   And 3 constraints:

   (a) $C_1(X, Y, Z)$ which is satisfied only when $X = Y + Z$

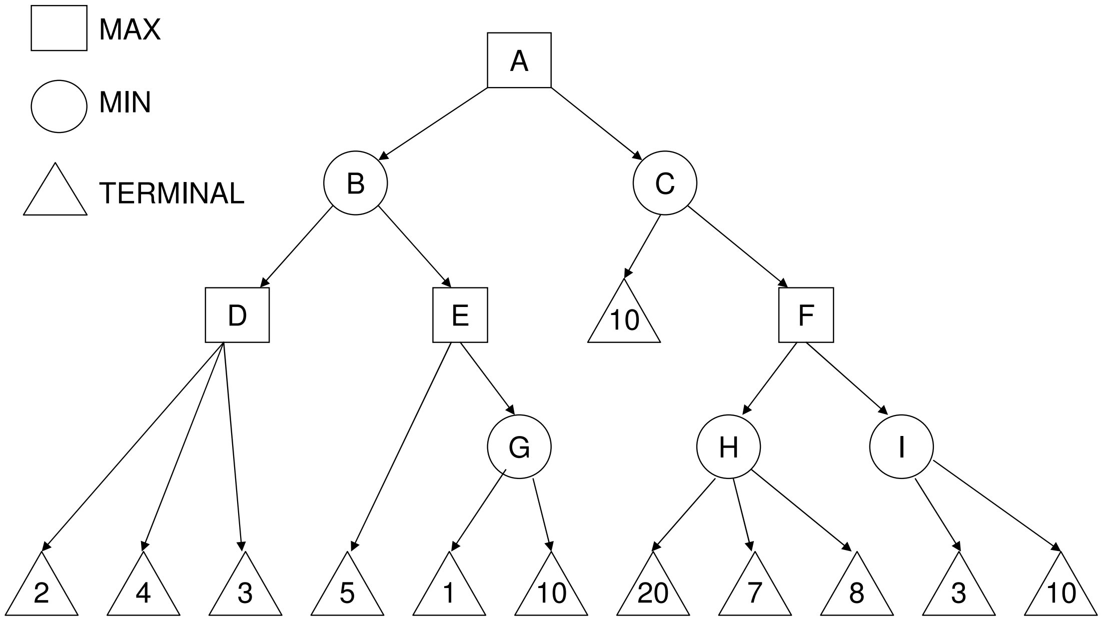
   (b) $C_2(X, W)$ which is satisfied only when $W > X$

   (c) $C_3(X, Y, Z, W)$ which is satisfied only when $W = X + Z + Y$

   Enforce GAC on these constraints, and give the resultant GAC consistent variable domains.

6 Game Tree Search

1. Consider the game tree shown below:



   (a) For each min and max node (A through I), list its minimax value in the space below:

A	B	C
D	E	F
G	H	I

   (b) On the diagram, circle the nodes that will be visited during a minmax depth-first search that uses *alpha-beta* pruning. You should also mark any terminal node that is visited. Draw a line across each edge that leads to a subtree pruned by an alpha or beta bound.

   (c) (1) What value does player MAX expect to get out of the game if MIN plays perfectly.

   (d) (1) What value does player MAX expect to get out of the game if MIN always moves to state E from state B.

   (e) (1) What value does player MAX expect to get out of the game if MIN always moves to state F from state C.