National University of Singapore
School of Computing
CS3243 — Foundations of Artificial Intelligence

Solutions for Mid-Term Test

March 13, 2008 **Time allowed:** 1 hour

Matriculation No: | | | | | | | | | |

Instructions (please read carefully):

1. Write down your matriculation number on the **question paper**. DO NOT WRITE YOUR NAME ON THE QUESTION SET!
2. You are allowed to bring one double-sided A4 sheet of notes for this test. Besides this sheet, you may not consult your books, friends, handphones and other reference materials for this test.
3. This paper comprises **THREE (3) questions** and **NINE (9) pages**. The time allowed for solving this quiz is **1 hour**.
4. The maximum score of this quiz is **60 marks**. The weight of each question is given in square brackets beside the question number.
5. All questions must be answered correctly for the maximum score to be attained.
6. All questions must be answered in the space provided in the answer sheet; no extra sheets will be accepted as answers.
7. The back-sides of the sheets and the pages marked "scratch paper" in the question set may be used as scratch paper.
8. You are allowed to use pencils, ball-pens or fountain pens, as you like (no red color, please).
9. The questions are presented in order by which their topic is covered in the syllabus, NOT by their perceived difficulty or estimated time to answer. You may wish to do the questions out of order.

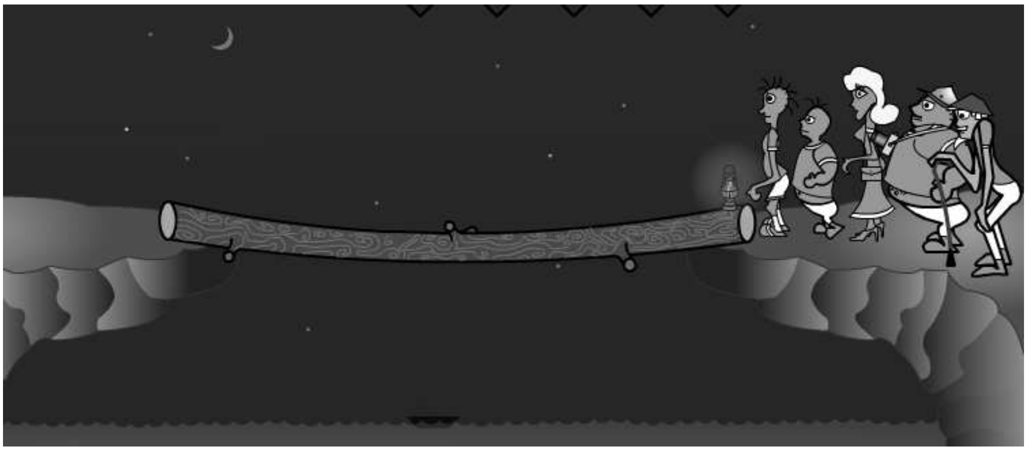
For Examiner's Use Only		
Question	Marks	Remark
Q1		
Q2		
Q3		
Total		

Question 1 : Crossing a Bridge in the Dark [24 marks]

A family of five were out on what they thought was a relaxing tour of the countryside when they became lost. They finally found a path that leads them back to civilization, but it is already pitch dark and they are confronted with a narrow and dangerous bridge.

They have only one lamp with them and only two persons can be on the bridge at any one time. Each person walks with a different speed and it will take 1, 3, 6, 8 and 12 seconds respectively for each of them to cross the bridge. Each pair must move at the pace of the slower person when they attempt to cross the bridge together.

There is only enough oil in the lamp to last 31 seconds. Can you help them? Fact: it is possible to cross the bridge within 31 seconds.



A. Propose a representation for the state of this game. State the initial and end states in your representation. [6 marks]

We can represent the state of the game as a tuple $(M_1, M_3, M_6, M_8, M_{12}, L, C)$*,* $D_{M_1} = D_{M_3} = D_{M_6} = D_{M_8} = D_{M_{12}} = D_L = \{L, R\}$*, where* M_1*,* M_3*,* M_6*,* M_8 *and* M_{12} *represent the position of the family members and L represents the position of the lamp. C is the total time taken (cost) to reach the state.* *[4 marks]*

Student will only be awarded one mark for the representation if C is missing.

Initial state: $(R, R, R, R, R, R, 0)$ *[1 marks]*

Goal state: (L, L, L, L, L, L, C), $C \leq 31$. *[1 marks]*

B. Suppose we want to find a way to cross the bridge within 31 seconds by applying TREE-SEARCH (See Appendix) using the following uninformed techniques:

1. Depth-first search (DFS)
2. Breath-first search (BFS)
3. Depth-limited search (DLS) (limited to depth 7)

4. Iterative-deepening search (IDS)

For each of these techniques, state if the search is (with respect to this particular problem):

1. complete? (yes/no) [hint: means will definitely find the solution]
2. optimal? (yes/no) [hint: means will definitely find the minimal solution]

Also, what are the expected orders of growth in time and space? For time and space complexities, let the maximum branching factor be b, the depth of the optimal solution be d and the maximum depth of the search tree be m. [8 marks]

This is a test of whether the student understands that the typical optimality and completeness conditions listed in the textbook are for unit-cost infinite search trees. In this question, the search tree is finite, so all the techniques are complete, including DLS at depth 7 since there is a solution at depth 7. The student is not required to come up with the actual solution, but from examining the problem, it should be obvious that there has to be a solution at depth 7, if there is a solution at all, which there has to be since we have specified that in the problem.

It turns out that there are at least two solutions that satisfy $C \leq 31$ at depth 7 where one is better than the other. Since we don't specify the order of expansion, it is possible for any of these algorithms to find either solution first and so none of them are optimal.

As for the complexities, they are by the book!

1. Depth-first search (DFS)
   - Complete: Yes
   - Optimal: No
   - Time Complexity: $O(b^m)$
   - Space Complexity: $O(bm)$
2. Breath-first search (BFS)
   - Complete: Yes
   - Optimal: No
   - Time Complexity: $O(b^{d+1})$
   - Space Complexity: $O(b^{d+1})$
3. Depth-limited search (DLS) (limited to depth 7)
   - Complete: Yes
   - Optimal: No
   - Time Complexity: $O(b^7)$

- Space Complexity: $O(b)$

4. Iterative-deepening search (IDS)

   - Complete: Yes
   - Optimal: No
   - Time Complexity: $O(b^d)$
   - Space Complexity: $O(bd)$

C. Assume (even if it's not true) at least one of the techniques employed in Part B is not complete, suggest a simple technique that can be applied to fix that, i.e. suggest how you would modify the search technique to make it complete. [2 marks]

Since the number of states is finite, remember all the states that have been visited before so as not to repeat, i.e. use GRAPH-SEARCH. *The purpose of this question is to test that students understand that a simple way to achieve completeness is to enumerate/visit all states.*

The family decides that uninformed search is too slow. You suggest improving the search by employing A* search and come up with the following three heuristics:

$$
\begin{aligned}
h_1(s) &= \text{Number of family members at the far side} \\
h_2(s) &= \text{Sum of times for the fastest half (rounded up) of the family members at the far side} \\
h_3(s) &= \text{Sum of times for all the family members at the far side}
\end{aligned}
$$

To clarify these two heuristics, assume we refer to each family member by the time he/she takes to cross the bridge. Suppose that in state s_1, 1 and 12 are on the near side with the lamp, while 3, 6 and 8 are on the far side. Then:

$$
\begin{aligned}
h_1(s_1) &= 3 \\
h_2(s_1) &= 3+6=9 \\
h_3(s_1) &= 3+6+8=17
\end{aligned}
$$

D. Are h_1, h_2 and h_3 admissible? Explain. If at least two of them are are admissible, compare their dominance. [8 marks]

Both h_1 and h_2 are admissible, while h_3 is not admissible.

For h_1, we enumerate the possibilities: (i) one person on far side; (ii) 1+3 on far side; (iii) all other cases and find that $h_1 \leq h^*$. *[2 marks]*

For h_2, it is the solution to the relaxed problem where there is no need for someone to bring the lamp back to the far side. *[2 marks]*

To see that h_3 is not admissible, consider the case where all except two family members are at the far side. Time taken will be the slower of the two, not the sum. *[2 marks]*

Between h_1 and h_2, neither are dominant. Consider two situations:

1. *1 and 3 on far side, then $h_1 = 2$ and $h_2 = 1$, so $h_1 > h_2$.*
2. *all 5 people on far side, then $h_1 = 5$ and $h_2 = 1+3+6 = 10$, so $h_2 > h_1$.*

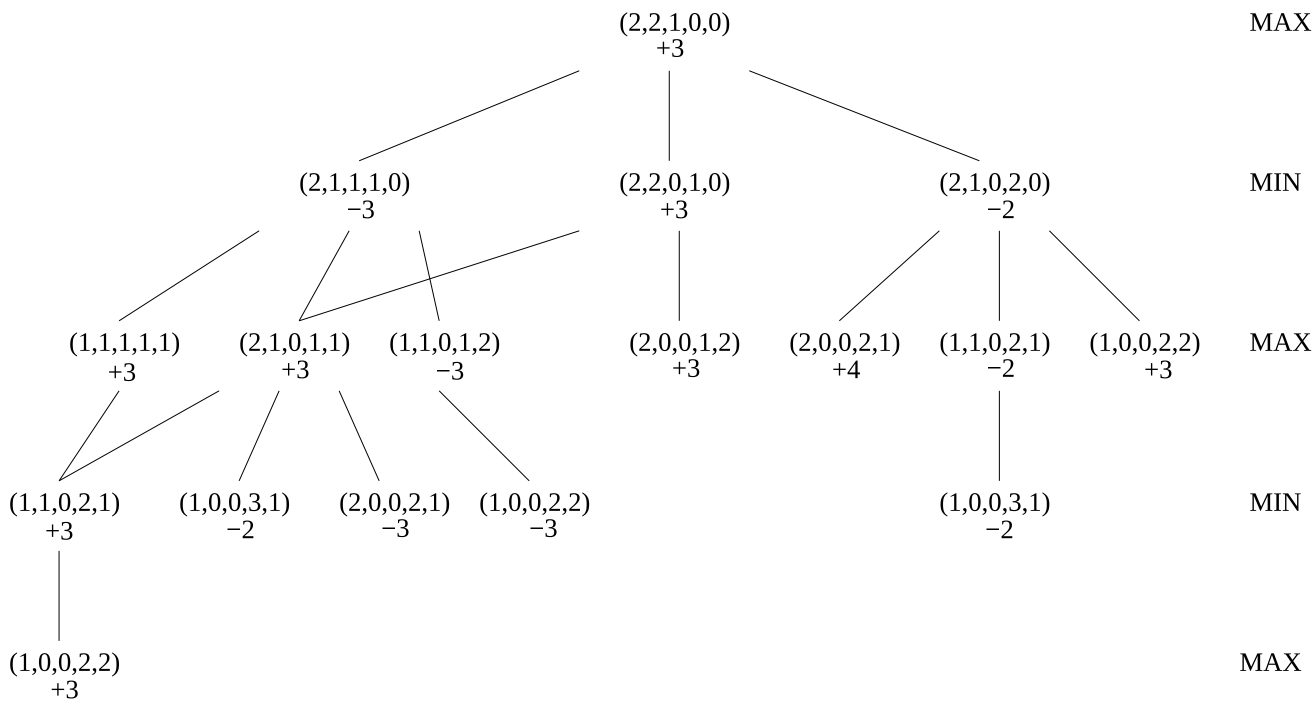
[2 marks]

Question 2 : Non-Optimal Opponents [21 marks]

In lecture, we discussed the Game of Nim with five sticks organized in three piles. In this problem, we consider a **variant** of the game with the same five sticks that has the following rules:

- Each player may **take only either one or two sticks** from an existing pile during his turn.
- The player who picks the last stick **wins** the game.
- In addition to winning, each player also tries pick up the maximum number of sticks.

We represent the game state as a tuple (p,q,r,s_1,s_2), $p \geq q \geq r$, where p, q and r are the number of sticks in the three piles and s_1 and s_2 are the number of sticks picked by the first and second players respectively. The following is the game tree for the game:



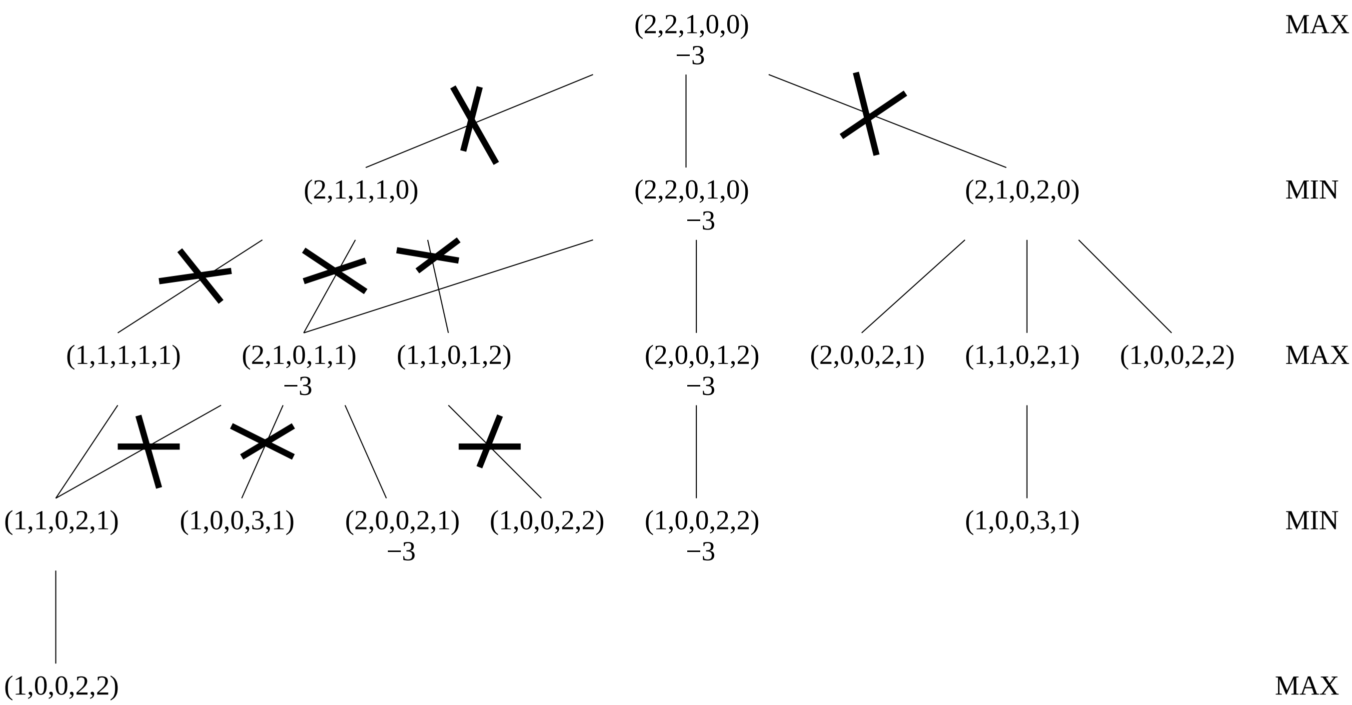
A. Let a state that results in a win for the first player (MAX) be of value $+s_1$, where s_1 is the number of sticks he picked. In a state that results in a win for the second player (MIN) the corresponding value is $-s_2$, where s_2 is the number of sticks picked by the second player. Solve the game by assigning a value to each node in the game tree above. If your objective is to win this game with the maximum number of sticks, would you opt to move first or second? [7 marks]

This is a standard question to test that the student understands how to compute minimax for a standard game tree.

5 marks for correct values for the nodes.

2 marks for stating we should start first, thereby demonstrating that the student can interpret the results correctly. Marks only awarded if the correct values are obtained for the game tree above (otherwise student might just be guessing and no marks should be awarded for random guesses).

B. [**Sub-Optimal Opponent**] Suppose you are playing against an opponent who always moves first, but who always picks one stick from the smallest pile. Solve the game by assigning values to the nodes in the game tree below



Can you win? [7 marks]

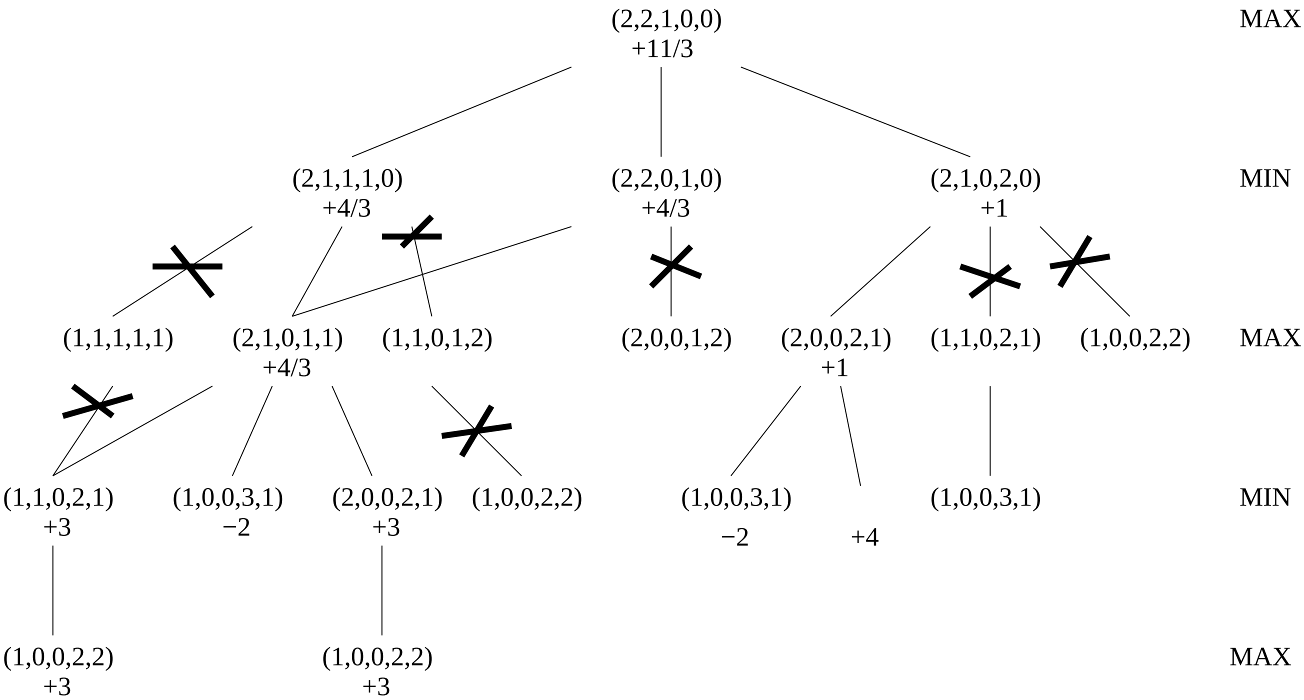
2 marks for pruning off the correct branches in the search tree.

3 marks for adding an extra level where necessary and computing the correct values.

2 marks for stating that the second player will ALWAYS win. Again, this statement must be supported by the correct game tree and it should not be a random guess.

C. [**Playing Optimally with Constraints and Uncertainty**] Suppose you are playing against an opponent who adopts the following strategy: among all the possible next moves, pick one uniformly at random. But before you rejoice, you have been struck by a bout of fever and cannot think straight: so during your turn, you can only pick one stick from the smallest pile.

It should be clear from the previous question that if we move first, we are sure to lose, so we only need to consider the case where we move second.



Would you opt to move first or second now? What's your probability of winning? Solve the game by assigning values to the nodes in the game tree above. [7 marks]

1 marks for pruning off the correct branches in the search tree.

2 marks for adding an extra level where necessary and computing the correct values.

2 marks for obeserving that we should always move second.

2 marks for computing the correct winning probability, which is: $\frac{1}{3} \cdot \frac{2}{3} + \frac{1}{2} \cdot \frac{1}{3} = \frac{7}{18}$.

Question 3 : Sudoku Anyone? [15 marks]

Sudoku is a logic-based number placement puzzle. The objective is to fill a 9×9 grid so that each column, each row, and each of the nine 3×3 boxes (also called blocks or regions) contains the digits from 1 to 9, only one time each (that is, exclusively). The puzzle setter provides a partially completed grid. *(Wikipedia)*

In this question, you will work with the following puzzle[1].

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

A. Describe how you would formulate the problem of solving the above puzzle as a backtracking search problem. In particular, explain also how you would apply the most constrained variable and least-constraining-value heuristics. [6 marks]

Assign a variable to each blank. For each variable, decide on its domain by considering 3 contraints – column, row and 3×3 *box.* *[2 marks]*

The most constrained variable is the variable with the smallest domain. To obtain the domain of a variable, we start with the full set from 0 to 9 and eliminate digits that already appear in its column, row and 3×3 *box.* *[2 marks]*

Once the variable is chosen, the least-constraining-value is the value that reduces the domain of the smallest number of unassigned variables. In other words, for each value in the domain of the unassigned variable, we count the number of **unassigned** *variables in the same column, row and* 3×3 *box for which the value has already been eliminated. We pick the value that is maximally eliminated from these unassigned variables.* *[2 marks]*

B. Can we apply AC-3 to the backtracking search you formulated in Part A above? Explain. [2 marks]

No. AC-3 is for arc consistency, i.e. binary constraints.

Yes, if you argue that we can convert the higher-order constraints to binary constraints.

[1] Note that the solution for the specified puzzle is irrelevant to this question. However, you are welcome to solve the puzzle if you finish this exam early for one bonus point. :-)

C. Describe how you would formulate this problem as a local search problem. In particular, describe (i) how do you come up with an initial state; (ii) how would you decide on the set of possible states in each subsequent local search step; and (iii) what is the heuristic you will employ to decide on the next state for each search step. [7 marks]

There are many possible answers to this question. This is a test that the student is able to formulate a CSP as a local search problem. Marks will be awarded for any <u>*reasonable*</u> *formulation.*

As before, assign a variable to each blank.

To obtain a start state, we can work on each 3×3 *box in turn. Randomly assign values to each of the unassigned variables such that the digits 1 to 9 all appear but once. This is the start state. [2 marks]*

Next we need a strategy to generate some next states. First, we count the number of constraints violations by each variable. Note that since each 3×3 *box contains only one of each digit, the only violations are in the row and columns. [2 marks]*

We use the min-conflicts heuristic. [1 marks]

Pick the variable that violates the most constraints as the target variable. The next step is generated by swapping this target variable with the values of one other variable within the same 3×3 *box. The next step is the configuration that reduces the total number of violated constraints globally. If the variable picked original cannot possibly achieve a reduction in the number of violated constraints with a swap, we try another variable and keep going until we find one that does reduce the number of violated constraints. [2 marks]*

Solution for puzzle:

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9