**CS 401 ARTIFICIAL INTELLIGENCE SECTION A**

**FAST, LAHORE CAMPUS, MIDTERM 1 EXAM**

MIDTERM 1 SOLUTIONS

**OBJECTIVE PART (MARKS 20)**

**TRUE / FALSE**

1. Minmax search will take less time when alpha beta pruning is not used ( F )
2. Hill climbing search is complete ( F )
3. Greedy best first search is conducted without the use of heuristics ( F )
4. The time complexity of A\* search is exponential ( T )
5. The order of expansion of nodes in iterative deepening search is the same as that of breadth first search ( F )
6. Consistent heuristics are always admissible heuristics ( T )
7. In Best-first search, when the list of open nodes is maintained as a sorted list, it is often referred to as a priority queue. ( T )
8. Heuristics that find the shortest path to a goal whenever it exists are said to be admissible. ( T )
9. Branching factor is a measure to determine the complexity of the search space. ( T )
10. In Depth-first search algorithm, the list of open nodes is maintained as a stack. ( T )

**Multiple Choice Questions: Circle the correct answer**

1. Which of the searches below is complete

1. Iterative deepening search with tree search
2. Depth first search with tree search
3. Greedy best first search with admissible heuristics
4. None of the above

2. Suppose you are searching in a grid of size *m*x*n* with *a* actions allowed. Breadth first search with tree search will have a time complexity of:

1. O(amn)
2. O(man)
3. O(amn)
4. O((mn)a)

3. Suppose you are searching in a grid of size *m*x*n* with *a* actions allowed. Depth first search with graph search will have a space complexity of:

1. O(amn)
2. O(man)
3. O(amn)
4. O((mn)a)

4. Which of the searches below is optimal in case all actions have a cost equal to 100

a. Uniform cost search with graph search

b. Breadth first search with tree search

c. A\* search with admissible heuristics and graph search

d. All of the above

5. In a state space graph, the ------------- of the graph represent legal moves from one board configuration to another.

1. root
2. **edges**
3. leaf nodes
4. None of the above

6. Artificial Intelligence is the ability to:

1. Perceive inter-relationship of facts
2. Learn and understand from experience
3. Acquire and retain knowledge
4. Respond quickly and successfully to a new situation
5. All of the above

7. The space complexity of minmax search is:

a. linear in terms of the maximum depth of the tree

b. exponential in terms of the maximum depth of the tree

c. logarithmic in terms of the maximum depth of the tree

d. none of the above. It depends upon the game

8. A necessary condition for the optimality of A\* search with tree search is:

a. All actions have the same cost and heuristics are consistent

b. All actions have the same cost and heuristics are admissible

c. All actions have different costs and heuristics are admissible

d. All of the above

9. Suppose you are searching in a grid where the following actions are allowed: diagonal up right, diagonal up left, right, left, up, down. Here the cost of each action is 1.5. The following heuristics are admissible:

a. Manhattan distance from state to goal

b. Euclidean distance from state to goal

c. Square of Euclidean distance

d. all of the above

10. Turing test approach says that a machine is intelligent if:

a. It can perform simple arithmetic calculations

b. It can fool a person into believing that he/she is talking to a human

c. It can beat a person at playing chess

d. None of the above

**QUESTION 2** Given the initial configuration of Sudoku game:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 1 | 3 | 2 |  |  | 3 | 2 | 1 |
| 1 | 2 | 4 | 3 |  | 3 |  | 1 | 4 |
| 1 | 3 | 2 | 4 |  | 2 | 1 |  | 3 |
| 3 | 4 | 1 | 2 |  | 1 |  | 3 | 2 |

Invalid configuration initial configuration

1. What is the heuristic function for your game? 2. Compute the heuristic function for the initial state.

3. Apply A\* search (graph search) to solve the above. Build the search tree and clearly indicate the order in which each state is expanded.

**SOLUTION: NOTE: This is just one possible solution. You can come up with a much better solution**

A possible heuristic function could be total columns with duplicates + total rows with duplicates+total empty cells. Order of exploration is shown as (a),(b),(c), (d)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3 | 2 | 1 |
| 3 |  | 1 | 4 |
| 2 | 1 |  | 3 |
|  |  |  |  |
| 1 |  | 3 | 2 |

f(n) = 0+3=3

(a)

Goal

f(n) = 2+1=3 (c)

f(n) = 1+2=3 (b)

f(n) = 3+2=5

f(n) = 3+1=4

f(n) = 3 (d)

f(n) = 3+2=5

f(n) = 2+2=4

f(n) = 2+3=5

f(n) = 2+3=5

f(n) = 1+4=5

f(n) = 1+4=5

f(n) = 1+3=4

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 2 | 1 | 4 |
| 2 | 1 | 2 | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 2 | 1 | 4 |
| 2 | 1 | 1 | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 2 | 1 | 4 |
| 2 | 1 | 4 | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 2 | 1 | 4 |
| 2 | 1 | 3 | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 2 | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 1 | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 4 | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 | 3 | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 3 | 2 | 1 |
| 3 |  | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 3 | 2 | 1 |
| 3 |  | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 3 | 2 | 1 |
| 3 |  | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 3 | 2 | 1 |
| 3 |  | 1 | 4 |
| 2 | 1 |  | 3 |
| 1 |  | 3 | 2 |

**QUESTION 3 (Marks 10)**

Use alpha beta pruning to find out which branches of the game tree are explored. If a branch is not explored then **cross it out**. If a branch is explored then put a **tick mark** against it. Don’t leave it blank. Also, show the values of alpha and beta wherever necessary. Alpha is the score for max and beta is the score for min. It is Min’s turn to play at the root level.

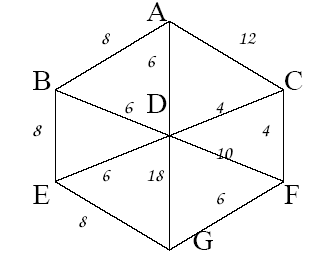


**QUESTION 4 (Marks 10)**

The numbers in the graph below are the real distances between the nodes. The estimated distances to the goal node, G, are the following:

A: 18, B: 13, C: 8, D: 18, E: 5, F: 6

A is the start node.



Implement a search with algorithm greedy best first search using graph search. Show how this algorithm works. Mark also in the tree the order in which the nodes are expanded.

SOLUTION: Order of expansion is A,C,F,G

**A**

f(A) = 18

**G**

f(G) = 8

**F**

f(F) = 8

**B**

f(B) = 13

**D**

f(D) = 18

**C**

f(C) = 8

**QUESTION 5 (Marks 4+4+2)**

Consider a game of Tic Tac Toe. **x** = Max, **o**=Min. The current board position of the game is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **x** | **o** |  | a.Make an exhaustive game tree from this configuration onwards. Its max’s turn to play.  b. Also use minmax algorithm to find the optimum score for max.  c. What is the likely outcome of the game and what is max’s next move? |
|  | **x** |  |  |
| **x** | **o** | **o** |  |

**SOLUTION**

|  |  |  |
| --- | --- | --- |
|  | **x** | **o** |
|  | **x** |  |
| **x** | **o** | **o** |

(0)

The utility of each game state is shown in the box below it. The game will end in a draw according to the minimax search algorithm. Max will choose the path shown by the bold line originating from the root.

(-1)

(-1)

(0)

(1)

(0)

(0)

(0)

(0)

(-1)

(0)

(1)

(-1)

(0)

|  |  |  |
| --- | --- | --- |
| o | **x** | **o** |
| x | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| o | **x** | **o** |
| x | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| x | **x** | **o** |
| o | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| x | **x** | **o** |
| o | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
|  | **x** | **o** |
| o | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| o | **x** | **o** |
|  | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
|  | **x** | **`** |
| x | **x** | o |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| o | **x** | **o** |
| x | **x** |  |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| x | **x** | **o** |
|  | **x** | o |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| x | **x** | **o** |
| o | **x** |  |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
|  | **x** | **o** |
|  | **x** | x |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
| x | **x** | **o** |
|  | **x** |  |
| **x** | **o** | **o** |

|  |  |  |
| --- | --- | --- |
|  | **x** | **o** |
| x | **x** |  |
| **x** | **o** | **o** |