82ABF456-BF5B-46ED-B8D3-E9DA8E08D8D6

1. Multiple Choice (10 points)

Circle the best choice for each question (2 points each):

1. What AI agent is most likely to have a natural language processing model?
   a. Deep Blue
   b. An agent in the DARPA Urban Challenge
   c. IBM Watson
   d. All of the above
   e. None of the above

2. Ignoring issues like traffic and speed limits, an admissible heuristic for route planning in Google Maps could be?
   a. Straight-line distance
   b. Manhattan distance
   c. h(n)=0
   d. Straight-line distance and Manhattan distance
   e. Straight-line distance and h(n)=0
   f. Straight-line distance, h(n)=0, and Manhattan distance

3. Which of the following is a contingency problem?
   a. Coloring the map of USA using 5 colors so that no 2 adjacent states have the same color
   b. Solving the 8 queens problem
   c. Finding a solution for the vacuum world problem where the agent is unreliable.
   d. Solving a Sudoku Puzzle
   e. None of the above

4. When searching a game tree, which search algorithm is used by the Minimax Algorithm (with or without alpha beta pruning) in absence of any dynamic move ordering scheme?
   a. A* search
   b. Breadth First Search
   c. Depth First Search
   d. Greedy Search
   e. Uniform Cost Search

5. The Imitation Game, mentioned in the paper by Alan Turing:
   a. overcomes the flaws in the Turing Test
   b. is an alternative to the Turing Test
   c. has several flaws
   d. All of the above
   e. None of the above

Q2 Score: **12.0**

2. True/False (20 points)

Circle True or False for each question (1 point each):

1. All of artificial intelligence research is about trying to pass Alan Turing's test, the Imitation Game. (True/False)
2. A rational agent can always achieve the best result. (True/False)
3. There exists a task environment in which every agent is rational. (True/False)
4. Depth First Search is a complete algorithm. (True/False)
5. When all costs are positive, Iterative Deepening Search will always find the optimal solution. (True/False)
6. Local beam search with one initial state and no limit on the number of states retained is Breadth First Search. (True/False)
7. Simulated annealing with T = 0 at all times is simple hill climbing. (True/False)
8. Mutation is a method used in the selection period of Genetic Algorithms. (True/False)
9. Cataclysm is a method to handle local extrema in Genetic Algorithms. (True/False)
10. If an agent can sense only partial information about the current state, it cannot behave rationally. (True/False)
11. If the entire state information is visible to a rational agent, it will select the optimal solution. (True/False)
12. From the perspective of an agent playing the game, blackjack is partially observable. (True/False)
13. The degree of pruning that occurs in an alpha-beta search is highly dependent on the order in which the states are examined. (True/False)
14. In an adversarial game playing agent using alpha-beta pruning, it would help search the game tree at a greater depth if the agent is using dynamic move-ordering schemes, for example, trying out moves that were found to be better in the past ahead of the ones found worse. (True/False)

2. True/False continued

15. Applying the Minimax algorithm to perform a full search of a game tree will give us different final results compared to using the Alpha Beta Pruning algorithm. (True/False)

16. The MAX player assumes that the MIN player will always play optimally. (True/False)

17. For chess, an appropriate evaluation function could be a linear weighted sum of features. (True/False)

18. Human champions refuse to compete against both Othello and Go playing agents, because computers are too good at both games. (True/False)

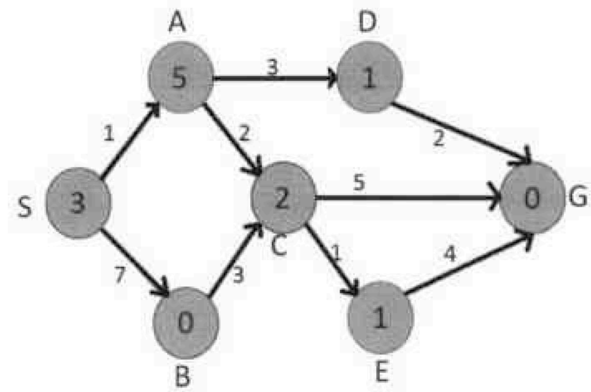
19. In AI, knowledge can make a big difference in solving a problem, but the particulars of how you structure the problem doesn't matter very much. (True/False)

20. The Recursive Best First Search (RBFS) algorithm was developed to deal with the time complexity problems of A* search. (True/False)

Q4 Score: **10.0**

3. Graph Search (20 points)

The following figure shows a graph. Each arc between nodes is labeled with the cost of traversing the link. Nodes are labeled with the heuristic estimate of the cost *h* of getting from the node to the goal. (Break the ties in alphabetical order)



1. In what order are nodes expanded by each of the following search algorithms, assuming each does loop check? (S = Start node, G = Goal node) (2 points each)

a. Depth-first search	SACEG
b. Breadth-first search	SABCDEG
c. Uniform-cost search	SACDEG
d. Greedy search	SBCG
e. A* search	SACDEG

DFS
G
E
G
C
D
A
B
S

BFS
S
A
B
C
D
E
G

UCS
S
1. A 1
B 7
2. C 3
3. D 4
4. E 4
5. G 8 6

Greedy
S
B
C
G

A*
S
1. A 6
B 7
2. C 5
3. D 5
G 8 6
4. E 5

S
1 A 1
B 7
2 C 3
3 D 4
4 E 4
5 G 8 6

Q5 Score: **5.0**

3. Graph Search continued

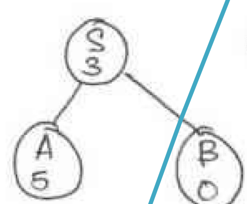
2. Can we decide whether h (as shown in the diagram) is an admissible heuristic function? Say why it is admissible or not and explain the reason. (5 points)

Yes we can decide that h is an admissible function because it does not over estimate the path cost and it gives the cost from that node to the goal. It gives a straight line distance so $h(n) < h^*(n) \leftarrow$ actual cost.

~~The heuristic~~

3. Can we decide whether it's a consistent heuristic function? Say whether or not it is and explain the reason. (5 points)

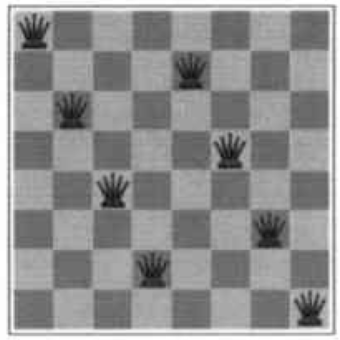
Yes we can decide because



according to this S will take the route to b all the time because the heuristic value of B is lesser than that of A. Because S take the path of B, we can say that $f(a) > f(b)$ and it is hence, consistent.

4. Local Search (10 points)

The problem is placing 8 queens on an 8 × 8 chessboard. The goal is that no two queens can attack each other, that is, that no two queens share the same row, column, or diagonal. In this question, we represent each state with an array of 8 elements, where the i(th) element is the row of the queen in the i(th) column. For the board shown it would be [1,3,5,7,2,4,6,8]. The action is to pick a queen and move it to another row in the same column.



1. How many neighbors states for this sample state? (3 points)

$^8P_8$

2. What is the total size of the search space, i.e. how many possible states are there in total? (3 points)

8!

3. Suppose we use a heuristic function h(n) that is the number of pairs of queens that are attacking each other. What is the h(n) of sample state shown? (2 points)

h(n) = 0

4. Give an example of a valid next state of hill-climbing algorithm or explain why there is no valid next state. (2 points)

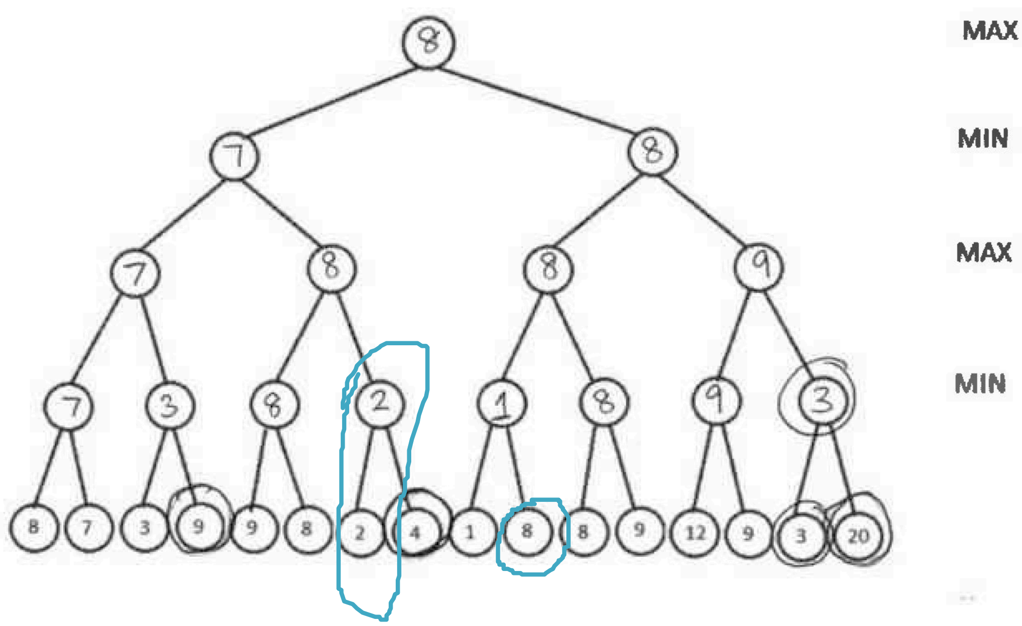
A valid next state may be to randomize and start from any point and start the algorithm again. This is random-restart.

Q7 Score: **8.5**

5. Game Playing (10 points)

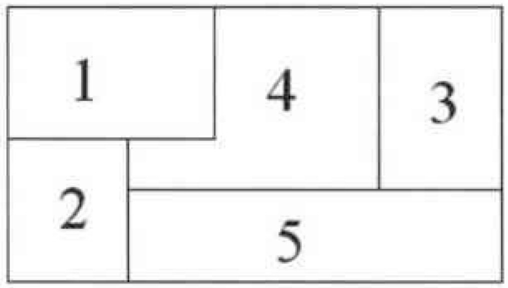
1. Using the Minimax algorithm, write the backed up values for each node (place the value in the circle). (5 points)

2. Circle the nodes (or subtrees) that would be pruned by alpha beta. (5 points)



6. Constraint Satisfaction (30 Points)

In this question, you will be solving a map-coloring problem. In this domain, each of the regions on a map must be colored black, gray, or red. The regions must be colored in such a way that no adjacent regions are the same color. Two regions are considered to be adjacent if they touch along an edge. Here is a map that you'll be using for all of the questions in this section:



1. Formulate this problem as a CSP problem with variables, domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit. (7 points)

Here the variables are the regions of the maps (it is ~~a~~ numbered 1 to 5) Hence we can say Variables = {1,2,3,4,5} and the values these variables can take are the colors black, gray, or red so domain = {black, gray, red}.

Variables = {1,2,3,4}

Domain for 1 = {black, red, gray}

Domain for 2 = —"—

Domain for 3 = —"—

Domain for 4 = {black, red, gray}

Domain for 5 = —"—

The constraint is that ① val(1) ≠ val(2), ≠ val(4) ② val(2), ≠ val(1), ≠ val(4) ③ val(4) ≠ val(1) ≠ val(2), ≠ val(5) ④ val(5) ≠ val(2), ≠ val(5), ≠ val(3)

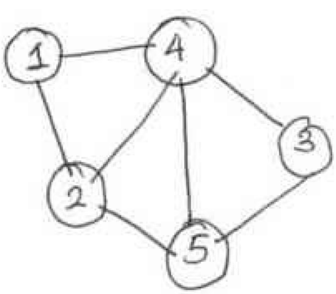
① Val1 cannot equal Val2 or Val4

② Val2 cannot equal Val1 or Val4

③ Val4 cannot equal Val1, Val2, Val5

④ Val5 cannot equal Val2, Val5, Val3

2. Draw the constraint graph associated with your CSP. (3 points)



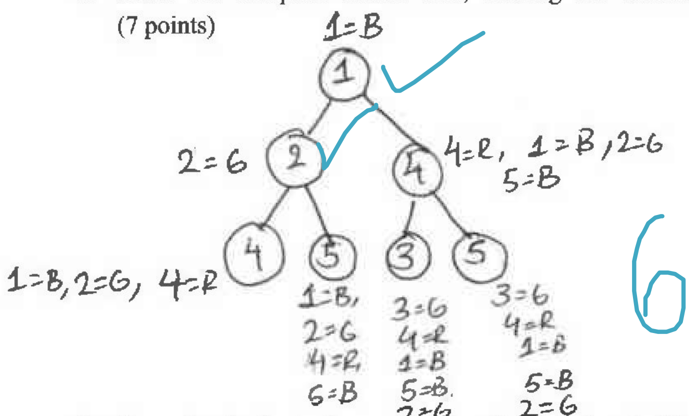
Nodes represnt variables and arcs represent constraints.

Q9 Score: **8.0**

6. Constraint Satisfaction continued

3. **Backtracking search**. Suppose we decide to use simple backtracking search to find a solution to this constraint satisfaction problem. The variable ordering heuristic we use is simply to instantiate the variables in numbered order. The value ordering heuristic is to consider the values in the order black, gray, red. You can use any reasonable shorthand to indicate the instantiations (e.g. "1=B" can mean region 1 is instantiated to the color black).

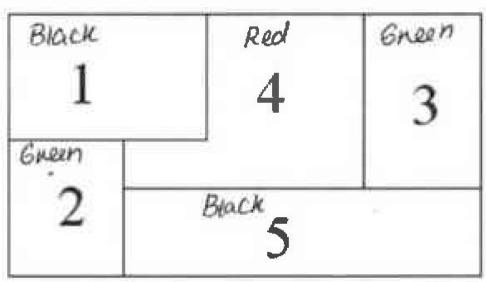
a. Show the complete search tree, circling the solution node, if one is found. (7 points)



B(black)
G(gray)
R(red)

ordering not followed -1

b. Show the final coloring on the map, if one is found. (2 points)



c. How many variable instantiations (search steps) are tried by this search method? (1 point)

5 (excluding the last leaf node)

6. Constraint Satisfaction continued

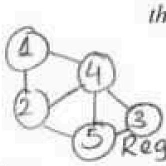
4. **Forward checking** Now suppose we use forward checking to eliminate illegal values from the domains of un-instantiated variables. Use the AC-3 algorithm. Furthermore, suppose we use a variable ordering heuristic that chooses the variable with the *fewest legal instantiations* remaining to instantiate next. If more than one such variable exist, the one earlier in the numbered order is selected. The same value ordering heuristic is used as in backtracking search (i.e., consider first black, then gray, then red).

a. Show the complete search table for forward checking search. At each node, *show the remaining legal values for the un-instantiated variables.* (7 points)

1st step | 2nd step | 5th step | 3rd step | 4th step

| B G R | ~~B~~ G ~~R~~ | ~~B~~ G ~~R~~ | ~~B~~ ~~G~~ R | B ~~G~~ ~~R~~ |

1 2 3 4 5



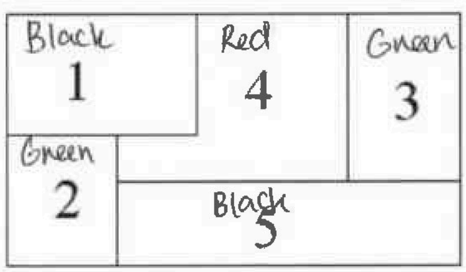
Explaination of Cutting Order:

Table ⇒

Region	Domain
1	B, ~~G~~, ~~R~~
2	~~B~~, G, ~~R~~
3	~~B~~, G, ~~R~~
4	~~B~~, ~~G~~, R
5	B, ~~G~~, ~~R~~

This cuts black from 2 and 4, choosing B for 1.
Acc to heuristic we choose color for 2 next. We choose gray acc. to the order. This cuts out gray for 4 and 5. Now 4 has least legal values left and is assigned red. This cuts out red for 3 and 5. 5 has only black remaining so that gets black and 3 gets green in the last move.

b. Show the final coloring on the map, if one is found. (2 points)



c. How many variable instantiations (search steps) are tried by this search method? (1 point)

5