Homework 02

Submission Notices:

- *Conduct your homework by filling answers into the placeholders in this file (in Microsoft Word format). Questions are shown in black color, instructions/hints are shown in italics and blue color, and your content should use any color that is different from those.*
- *After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,*
  <StudentID-1>_<StudentID-2>_HW01.pdf (Student IDs are sorted in ascending order)
  E.g., ***2112001_2112002_HW02.pdf***
  and then submit the file to Moodle directly WITHOUT any kinds of compression (.zip, .rar, .tar, etc.).
- *Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.*
  1. *Wrong file/filename format, e.g., not a pdf file, use "-" instead of "_" for separators, etc.*
  2. *Disorder format of problems and answers*
  3. *Conducted not in English*
  4. *Cheating, i.e., copying other students' works or letting other students copy your work.*

Problem 1. (1pt) Answer the following simple questions.

Please write your answer in the table

Questions (0.25pt each)	***Filling in the blanks***
What is local search in constraint satisfaction?	*Local search is a heuristic method for solving computationally hard optimization problems.*
What is a ridge in the local search algorithm?	*A ridge is a sequence of local maxima where every neighbor appears to be downhill but the search space has an uphill.*
What is the objective of the greedy descent local search in CSPs by applying the heuristic function?	*The goal is to find a solution to the problem by iteratively improving an initial assignment of the variables until all constraints are satisfied. The heuristic function is used to determine which variable to modify at each step.*
What is the main drawback of the Hill climbing algorithm, and how can it be addressed?	*The answer is that it can get stuck in local maxima or plateaus and fail to find the global maximum. One way to address this issue is to use random restarts where the algorithm is restarted from a random initial state after it gets stuck in a local maximum.*

Problem 2. (1pt) For each of the following question, please choose either True or False and give a brief explanation.

Please write your answer in the table

Claims	**True/False**	**Explanation**
Hill climbing can be used for both optimization and search problems.	T	Hill climbing is a simple optimization algorithm that starts with an initial solution and iteratively improves it by moving to a better neighboring solution. It can be used for both optimization and search problems because it can be applied to any problem that can be represented as a state space search problem.
It is possible to get stuck in a local maximum in simulated annealing.	T	It is possible to get stuck in a local maximum in simulated annealing because it relies on finding lucky jumps that improve the position. If the algorithm gets stuck in a local maximum, it will not be able to find the global maximum.

Problem 3. (1pt) *8-puzzle problem*. Apply the *hill-climbing algorithm* with the *sum of Manhattan distance heuristic* to find a solution for the following pair of initial and goal states.

Initial state:

1	2	3
7		6
5	4	8

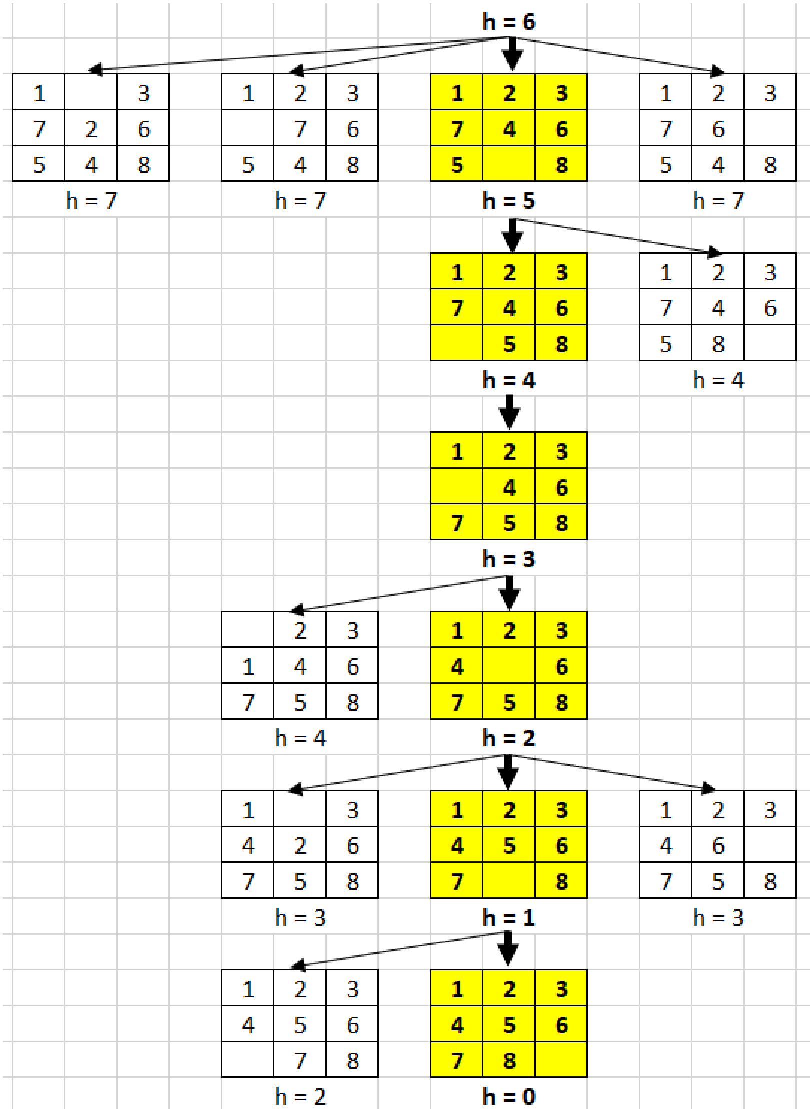
Goal state:

1	2	3
4	5	6
7	8	

Your work should address the following requirements

- Draw the search tree including all possible successors of expanded states (except the goal)
- Calculate the heuristic value for every node
- Mark the optimal strategy found

Note that there will be ties at some steps. The hill-climbing search may be stuck after a few moves if it follows an unpromising branch, and unfortunately, it has no way to predict this. Let's assume that our search is "smart" enough to pick the correct one among ties.



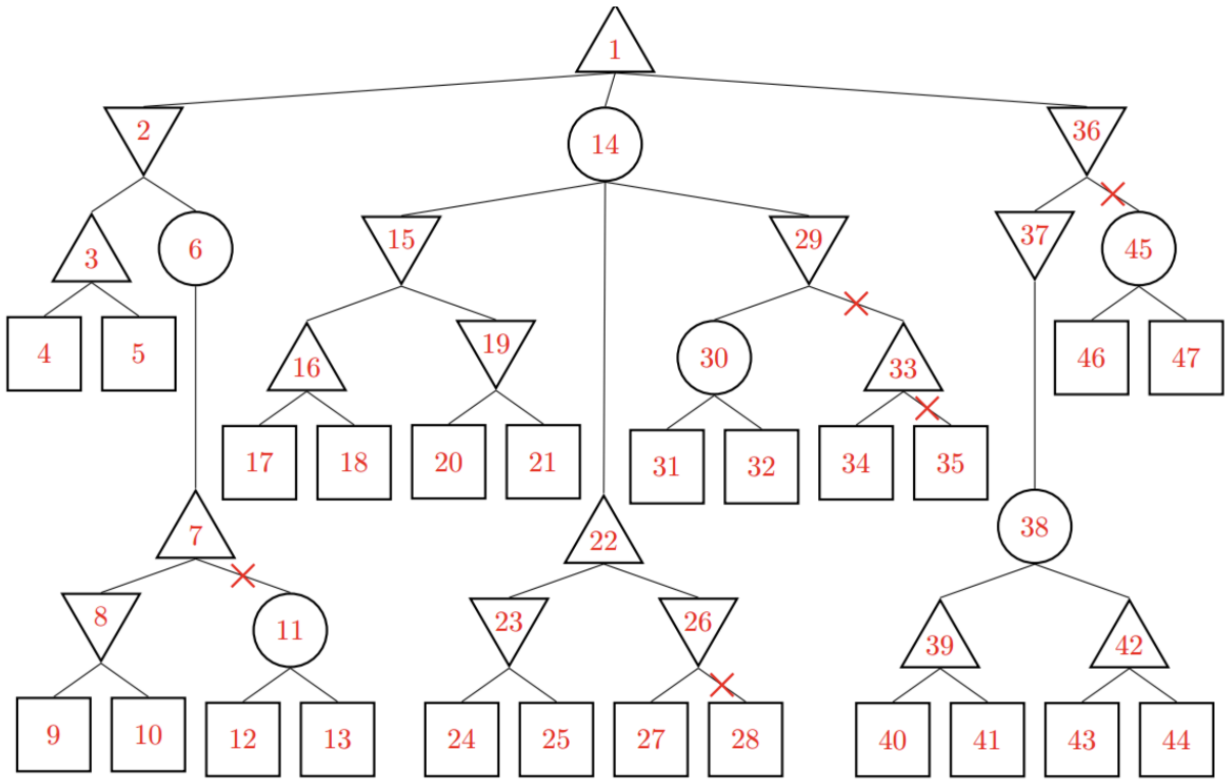
Each incorrect/missing state gives -0.125pt. No optimal strategy marked: -0.25pt.
States on the optimal path are marked in yellow color and bolded font.

Problem 4. (3pts) Given the game tree below, note that:

- The max nodes are denoted as upward pointing triangles (Δ).
- The min nodes are denoted as downward pointing triangles (∇).
- The chance nodes are denoted as circles (⊙).
- The square denotes the leaf nodes (⊡).

a) (1.5pts) Assuming that the children of a node are visited in left-to-right order, put an 'X' on the branches if there exists an assignment of values to leaf nodes, for which that branch could be pruned. If no branches can be pruned, please write down "not possible".

b) (1.5pts) State why these branches are chosen in case they exist.

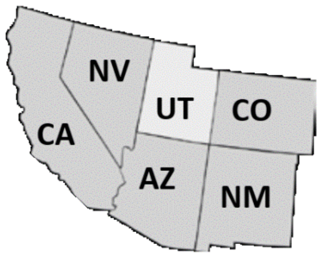


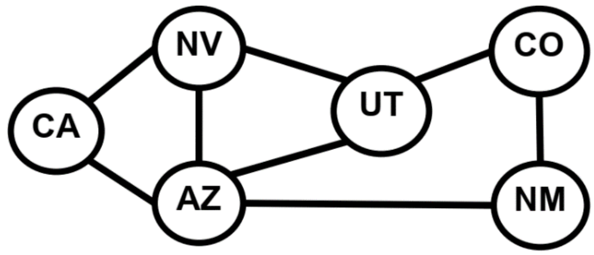
Please write your answer in the table

When there is an adversarial min-max set of nodes, then bounds can be imposed on how good or bad a node can be which allows us to prune certain branches. However, we always must look at the left most branch in order to determine whether pruning is ever going to be possible. Branches immediately under a chance node, however, can never be pruned because they all need to be looked at to compute an expectation. *The numbers inside the nodes have been added to the solution for the sole purpose of referencing them in the explanations that follow. There are 5 branches that can be pruned (from left to right):* *The first branch (the one above node 11) can be pruned if the minimizer node 8 has a value that is greater than the value that the minimizer node 2 has so far (i.e., the value of node 3). This is what typically happens in a standard minimax tree since the chance node 6 with a single child can be thought of as not doing anything.* *The second branch (the one above leaves 28) can be pruned if the leave node 27 has a value that is less than the value that the maximizer node 22 has so far (i.e. the value of the node 23). The third branch can be pruned (the one above node 33) if the chance node 14 has a value that is, so far, less than the value that the root maximizer node 1 has so far. This is because once the value of the chance node 30 is known, the value of the chance node 14 can't increase.* *The fourth branch (the one above leaves 35) can be pruned if the leaf 34 has a value that is greater than the value of the chance node 30. Even though the third branch can be pruned, this fourth one should also be marked because there might be cases in which the third branch can't be pruned but the fourth branch can.* *However, the rubric doesn't penalize for not marking the fourth branch if the third branch is marked.* *This is because the question didn't explicitly specify to mark branches that are underneath branches that can be pruned.*

The fifth branch (the one above the chance node 45) can be pruned if the minimizer node 37 has a value that is less than the value that the root maximizer node 1 has so far.

Problem 5. (2pts) Given *Constraint Satisfaction Problem*. You are a map-coloring robot assigned to color this Southwest USA map. Adjacent regions must be colored a different color (R=Red, B=Blue, G=Green). The constraint graph is shown below.





Note that the following questions are mutually independent.

a) (0.5pt) Mark all values that would be eliminated by *Forward Checking*, after variable **AZ** has just been assigned value R as shown.

CA	NV	AZ	UT	CO	NM
R G B	**R** G B	**R**	**R** G B	R G B	**R** G B

b) (0.5pt) **CA** and **AZ** have been assigned values, but no constraint propagation has been done. Mark all values that would be eliminated by *Arc Consistency AC-3*.

CA	NV	AZ	UT	CO	NM
B	**R** G **B**	**R**	**R** **G** B	R G **B**	**R** G B

c) (0.5pt) Consider the assignment below, where. **NV** is assigned and constraint propagation has been done. Circle all unassigned variables that might be selected by the *Minimum-Remaining-Values (MRV) Heuristic*.

CA	NV	**AZ**	**UT**	CO	NM
R B	**G**	R B	R B	R G B	R G B

d) (0.5pt) Consider the assignment below. **NV** is assigned and constraint propagation has been done. Circle all unassigned variables that might be selected by the *Degree Heuristic*.

CA	NV	**AZ**	UT	CO	NM
R B	**G**	R B	R B	R G B	R G B

Problem 6. (2pts) Alice, Betty, and Carol are in a book club. They're trying to figure which of five different books they should read next. The books are:

- (D): Dreams From My Father by Barack Obama, non-fiction
- (L): Lord of the Rings by J.R.R. Tolkein (L), fiction

- (A): Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, non-fiction
- (H) Harry Potter and The Sorcerer's Stone, by J.K. Rowling, fiction, and
- (F): The Fabric of the Cosmos: Space, Time, and the Texture of Reality by Brian Greene, non-fiction.

In this book club, they can't read the same book because there is only one copy for each book. Betty only reads non-fiction. Alice won't read whatever either Betty or Carol are reading, while Betty and Carol always read the same type of book.

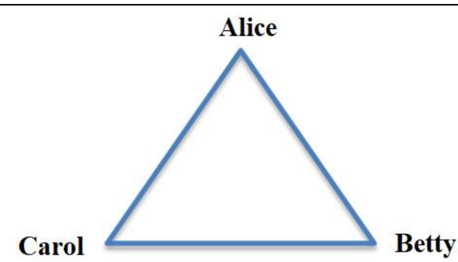
a) (1pt) Formulate this problem as a CSP, stating the variables, domains, and constraints. Constraints should be specified formally and precisely but may be implicit rather than explicit.

Please write your answer in the table

Variables	Alice	Betty	Carol
Initial domains (after enforcing unary constraints)	D,L,A,H,F	D, A, F	D,L,A,H,F
Binary constraints (each constraint is represented as a pair of variables participating in the relation	Alice - Betty Alice - Carol Betty - Carol		

b) (0.5pt) Draw the constraint graph associated with your CSP.

Please present your work in the table



c) (0.5pt) Assume that Betty asked you to choose a book for her. Starting from the domains in a), show the resulting domains after running arc-consistency.

Please write your answer in the table

It depends on which book you choose for Betty. The following solution demonstrates the case of choosing the book D for Betty.

Variables	Alice	Betty	Carol
Domains	L,A,H,F	**D**	A,F