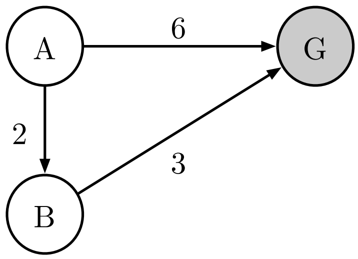
Q1. [6 pts] Search: Heuristic Function Properties

For the following questions, consider the search problem shown on the left. It has only three states, and three directed edges. A is the start node and G is the goal node. To the right, four different heuristic functions are defined, numbered I through IV.



	h(A)	h(B)	h(G)
I	4	1	0
II	5	4	0
III	4	3	0
IV	5	2	0

(a) [4 pts] **Admissibility and Consistency**

For each heuristic function, circle whether it is admissible and whether it is consistent with respect to the search problem given above.

	Admissible?		Consistent?	
I	$\boxed{\text{Yes}}$	No	Yes	$\boxed{\text{No}}$
II	Yes	$\boxed{\text{No}}$	Yes	$\boxed{\text{No}}$
III	$\boxed{\text{Yes}}$	No	$\boxed{\text{Yes}}$	No
IV	$\boxed{\text{Yes}}$	No	Yes	$\boxed{\text{No}}$

II is the only inadmissible heuristic, as it overestimates the cost from B: $h(B) = 4$, when the actual cost to G is 3.

To check whether a heuristic is consistent, ensure that for all paths, $h(N) - h(L) \leq \text{path}(N \to L)$, where N and L stand in for the actual nodes. In this problem, $h(G)$ is always 0, so making sure that the direct paths to the goal ($A \to G$ and $B \to G$) are consistent is the same as making sure that the heuristic is admissible. The path from A to B is a different story.

Heuristic I is not consistent: $h(A) - h(B) = 4 - 1 = 3 \not\leq \text{path}(A \to B) = 2$.
Heuristic III is consistent: $h(A) - h(B) = 4 - 3 = 1 \leq 2$
Heuristic IV is not consistent: $h(A) - h(B) = 5 - 2 = 3 \not\leq 2$

(b) [2 pts] **Function Domination**

Recall that *domination* has a specific meaning when talking about heuristic functions.

Circle all true statements among the following.

1. Heuristic function III dominates IV.
2. Heuristic function IV dominates III.
3. $\boxed{\text{Heuristic functions III and IV have no dominance relationship.}}$
4. Heuristic function I dominates IV.
5. $\boxed{\text{Heuristic function IV dominates I.}}$
6. Heuristic functions I and IV have no dominance relationship.

For one heuristic to dominate another, *all* of its values must be greater than or equal to the corresponding values of the other heuristic. Simply make sure that this is the case. If it is not, the two heuristics have no dominance relationship.

Q2. [30 pts] Search problems

It is training day for Pacbabies, also known as Hungry Running Maze Games day. Each of k Pacbabies starts in its own assigned start location s_i in a large maze of size MxN and must return to its own Pacdad who is waiting patiently but proudly at g_i; along the way, the Pacbabies must, between them, eat all the dots in the maze.

At each step, all k Pacbabies move one unit to any open adjacent square. The only legal actions are Up, Down, Left, or Right. It is illegal for a Pacbaby to wait in a square, attempt to move into a wall, or attempt to occupy the same square as another Pacbaby. To set a record, the Pacbabies must find an optimal collective solution.

(a) [5 pts] Define a minimal state space representation for this problem.

The state space is defined by the current locations of k Pacbabies and, for each square, a Boolean variable indicating the presence of food.

(b) [2 pts] How large is the state space?

$(MN)^k \cdot 2^{MN}$

(c) [3 pts] What is the maximum branching factor for this problem?

(A) 4^k (B) 8^k (C) $4^k 2^{MN}$ (D) $4^k 2^4$

Each of k Pacbabies has a choice of 4 actions.

(d) [6 pts] Let $MH(p, q)$ be the Manhattan distance between positions p and q and F be the set of all positions of remaining food pellets and p_i be the current position of Pacbaby i.
Which of the following are admissible heuristics?

h_A: $\frac{\sum_{i=1}^{k} MH(p_i, g_i)}{k}$
h_B: $\max_{1 \le i \le k} MH(p_i, g_i)$

h_C: $\max_{1\leq i\leq k}[\max_{f\in F} MH(p_i, f)]$
h_D: $\max_{1\leq i\leq k}[\min_{f\in F} MH(p_i, f)]$
h_E: $\min_{1\leq i\leq k}[\min_{f\in F} MH(p_i, f)]$
h_F: $\min_{f\in F}[\max_{1\leq i\leq k} MH(p_i, f)]$

h_A is admissible because the total Pacbaby–Pacdad distance can be reduced by at most k at each time step.
h_B is admissible because it will take at least this many teps for the furthest Pacbaby to reach its Pacdad.
h_C is inadmissible because it looks at the distance from each Pacbaby to its most distant food square; but of course the optimal solution might another Pacbaby going to that square; same problem for h_D.
h_E is admissible because some Pacbaby will have to travel at least this far to eat one piece of food (but it's not very accurate).
h_F is inadmissible because it connects each food square to the most distant Pacbaby, which may not be the one who eats it.
A different heuristic, $h_G = \max_{f\in F}[\min_{1\leq i\leq k} MH(p_i, f)]$, *would* be admisible: it connects each food square to its closest Pacbaby and then considers the most difficult square for any Pacbaby to reach.

(e) [2 pts] Give one pair of heuristics h_i, h_j from part (d) such that their *maximum* — $h(n) = max(h_i(n), h_j(n))$ — is an admissible heuristic.

Any pair from h_A, h_B, and h_E: the max of two admissible heuristics is admissible.

(f) [2 pts] Is there a pair of heuristics h_i, h_j from part (d) such that their *convex combination* — $h(n) = \alpha h_i(n) + (1-\alpha)h_j(n))$ — is an admissible heuristic for any value of α between 0 and 1? Briefly explain your answer.

Any pair from h_A, h_B, and h_E: the convex combination of two admissible heuristics is dominated by the max, which is admissible.

Now suppose that some of the squares are flooded with water. In the flooded squares, it takes two timesteps to travel through the square, rather than one. However, the Pacbabies don't know which squares are flooded and which aren't, until they enter them. After a Pacbaby enters a flooded square, its howls of despair instantly inform all the other Pacbabies of this fact.

(g) [4 pts] Define a minimal space of belief states for this problem.

The physical states about which the agent is uncertain are configurations of MN wetness bits, of which there are 2^{MN}. In general, the space of belief states would be all possible subsets of the physical states, i.e., $2^{2^{MN}}$ subsets of the 2^{MN} configurations. However, percepts in this world give either no information about a location or perfect information, so the reachable belief states are those 3^{MN} belief states in which each square is wet, dry, or unknown. Either answer is OK.

(h) [2 pts] How many possible environmental configurations are there in the initial belief state, before the Pacbabies receive any wetness percepts?

2^{MN}

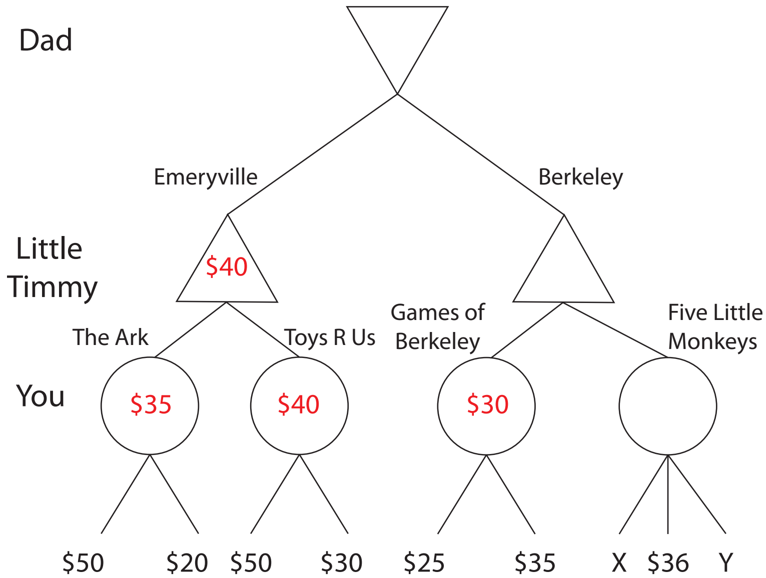
(i) [4 pts] Given the current belief state, how many different belief states can be reached in a single step?

(A) 4^k (B) 8^k (C) $4^k 2^{MN}$ (D) $4^k 2^4$

After each of 4^k joint movements of Pacbabies, there are 2^k possible joint percepts, each leading to a distinct belief state.

Q3. [9 pts] Expectimax

Your little brother Timmy has a birthday and he was promised a toy. However, Timmy has been misbehaving lately and Dad thinks he deserves the least expensive present. Timmy, of course, wants the most expensive toy. Dad will pick the city from which to buy the toy, Timmy will pick the store and you get to pick the toy itself. You don't want to take sides so you decide to pick a toy at random. All prices (including X and Y) are assumed to be nonnegative.



(a) [1 pt] Fill in the values of all the nodes that don't depend on X or Y.

(b) [3 pts] What values of X will make Dad pick Emeryville regardless of the price of Y?

Dad will pick Emeryville if the value of the Berkeley node is more than the value of the Emeryville node ($40), that is if:

$$\frac{x+y+36}{3} > 40 \Leftrightarrow x > 84 - y \Leftrightarrow x > 84$$

.

(c) [3 pts] We know that Y is at most $30. What values of X will result in a toy from Games of Berkeley regardless of the exact price of Y?

Games of Berkeley will be chosen if the value of the "Five Little Monkeys" node is less than the value of the "Games of Berkeley" node ($30):

$$\frac{x+y+36}{3} < 30 \Leftrightarrow x < 54 - y \Leftrightarrow x < 24$$

(d) [2 pts] Normally, alpha-beta pruning is not used with expectimax. However, with some additional information, it is possible to do something similar. Which **one** of the following conditions on a problem are required to perform pruning with expectimax?

1. The children of the expectation node are leaves.
2. All values are positive.
3. The children of the expectation node have specified ranges.
4. The child to prune is last.

The key observation is that any single child of an expectation node can make the value of the node arbitrarily small or large unless the value of the child is known to be within some specific range.

Q4. [17 pts] CSPs

(a) Pacman's new house

After years of struggling through mazes, Pacman has finally made peace with the ghosts, Blinky, Pinky, Inky, and Clyde, and invited them to live with him and Ms. Pacman. The move has forced Pacman to change the rooming assignments in his house, which has 6 rooms. He has decided to figure out the new assignments with a CSP in which the variables are Pacman **(P)**, Ms. Pacman **(M)**, Blinky **(B)**, Pinky **(K)**, Inky **(I)**, and Clyde **(C)**, the values are which room they will stay in, from 1-6, and the constraints are:

i) No two agents can stay in the same room
ii) **P** > 3
iii) **K** is less than **P**
iv) **M** is either 5 or 6
v) **P** $>$ **M**
vi) **B** is even
vii) **I** is not 1 or 6
viii) $|\mathbf{I}\text{-}\mathbf{C}| = 1$
ix) $|\mathbf{P}\text{-}\mathbf{B}| = 2$

(i) [1 pt] **Unary constraints** On the grid below cross out the values from each domain that are eliminated by enforcing unary constraints.

P	~~1~~	~~2~~	~~3~~	4	5	6
B	~~1~~	2	~~3~~	4	~~5~~	6
C	1	2	3	4	5	6
K	1	2	3	4	5	6
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	5	6

The unary constraints are ii, iv, vi, and vii. ii crosses out 1,2, and 3 for P. iv crosses out 1,2,3,4 for M. vi crosses out 1,3, and 5 for B. vii crosses out 1 and 6 for I. K and C have no unary constraints, so their domains remain the same.

(ii) [1 pt] **MRV** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned to first?

○ P ○ B ○ C ○ K ○ I ● M

M has the fewest value remaining in its domain (2), so it should be selected first for assignment.

(iii) [2 pts] **Forward Checking** For the purposes of decoupling this problem from your solution to the previous problem, assume we choose to assign P first, and assign it the value 6. What are the resulting domains after enforcing unary constraints (from part i) and running forward checking for this assignment?

P						6
B	~~1~~	~~2~~	~~3~~	4	~~5~~	~~6~~
C	1	2	3	4	5	~~6~~
K	1	2	3	4	5	~~6~~
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	5	~~6~~

In addition to enforcing the unary constraints from part i, the domains are further constrained by all constraints involving P. This includes constraints i, iii, v, and ix. i removes 6 from the domains of all variables. iii removes 6 from the domain of K (already removed by constraint i). v removes 6 from the domain of M (also already removed by i). ix removes 2 and 6 from the domain of B.

(iv) [3 pts] **Iterative Improvement** Instead of running backtracking search, you decide to start over and run iterative improvement with the min-conflicts heuristic for value selection. Starting with the following assignment:

P:6, B:4, C:3, K:2, I:1, M:5

First, for each variable write down how many constraints it violates in the table below.
Then, in the table on the right, for all variables that could be selected for assignment, put an x in any box that corresponds to a possible value that could be assigned to that variable according to min-conflicts. When marking next values a variable could take on, only mark values different from the current one.

Variable	# violated
P	0
B	0
C	1
K	0
I	2
M	0

	1	2	3	4	5	6
P						
B						
C		x				
K						
I		x		x		
M						

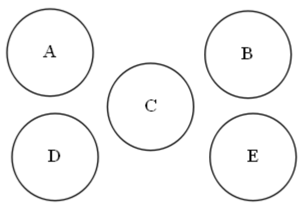
Both I and C violate constraint viii, because |I-C|=2. I also violates constraint vii. No other variables violate any constraints. According to iterative improvement, any conflicted variable could be selected for assignment, in this case I and C. According to min-conflicts, the values that those variables can take on are the values that minimize the number of constraints violated by the variable. Assigning 2 or 4 to I causes it to violate constraint i, because other variables already have the values 2 and 4. Assigning 2 to C also only causes C to violate 1 constraint.

(b) All Satisfying Assignments Now consider a modified CSP in which we wish to find every possible satisfying assignment, rather than just one such assignment as in normal CSPs. In order to solve this new problem, consider a new algorithm which is the same as the normal backtracking search algorithm, except that when it sees a solution, instead of returning it, the solution gets added to a list, and the algorithm backtracks. Once there are no variables remaining to backtrack on, the algorithm returns the list of solutions it has found.

For each graph below, select whether or not using the MRV and/or LCV heuristics could affect the number of nodes expanded in the search tree in this new situation.

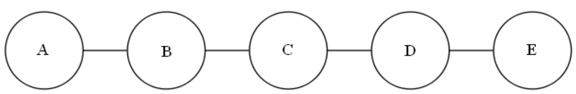
The remaining parts all have a similar reasoning. Since every value has to be checked regardless of the outcome of previous assignments, the order in which the values are checked does not matter, so LCV has no effect.
In the general case, in which there are constraints between variables, the size of each domain can vary based on the order in which variables are assigned, so MRV can still have an effect on the number of nodes expanded for the new "find all solutions" task.
The one time that MRV is guaranteed to not have any effect is when the constraint graph is completely disconnected, as is the case for part i. In this case, the domains of each variable do not depend on any other variable's assignment. Thus, the ordering of variables does not matter, and MRV cannot have any effect on the number of nodes expanded.

(i) [2 pts]



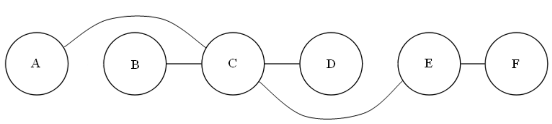
● Neither MRV nor LCV can have an effect.
○ Only MRV can have an effect.
○ Only LCV can have an effect .
○ Both MRV and LCV can have an effect.

(ii) [2 pts]



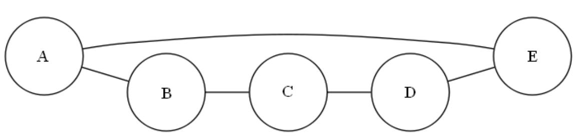
○ Neither MRV nor LCV can have an effect.
● Only MRV can have an effect.
○ Only LCV can have an effect .
○ Both MRV and LCV can have an effect.

(iii) [2 pts]



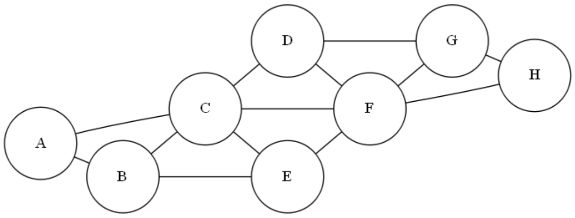
○ Neither MRV nor LCV can have an effect.
● Only MRV can have an effect.
○ Only LCV can have an effect .
○ Both MRV and LCV can have an effect.

(iv) [2 pts]



○ Neither MRV nor LCV can have an effect.
● Only MRV can have an effect.
○ Only LCV can have an effect .
○ Both MRV and LCV can have an effect.

(v) [2 pts]



- ○ Neither MRV nor LCV can have an effect.
- ● Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

Q5. [20 pts] Propositional logic

(a) [6 pts] Consider a vocabulary with only four symbols, A, B, C, and D. For each of the following sentences, how many possible worlds make it true?

1. $(A \wedge B) \vee (C \wedge D)$ 7 (4 for $A \wedge B$, 4 for $C \wedge D$, minus 1 for the model that satisfies both).

2. $\neg(A \wedge B \wedge C \wedge D)$ 15 — it's the negation of a sentence with 1 model.

3. $B \Rightarrow (A \wedge B)$ 12 — it's true when B is false (8) and when B is true and A is true (4).

(b) [8 pts] A certain procedure to convert a sentence to CNF contains four steps (1-4 below); each step is based on a logical equivalence. Circle ALL of the valid equivalences for each step.

1. Step 1: drop biconditionals

   $\boxed{a)}$ $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$

   b) $(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \vee (\beta \Rightarrow \alpha))$

   c) $(\alpha \Leftrightarrow \beta) \equiv (\alpha \wedge \beta)$

2. Step 2: drop implications

   a) $(\alpha \Rightarrow \beta) \equiv (\alpha \vee \neg\beta)$

   $\boxed{b)}$ $(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$

   c) $(\alpha \Rightarrow \beta) \equiv (\neg\alpha \wedge \beta)$

3. Step 3: move not inwards

   $\boxed{a)}$ $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$

   b) $\neg(\alpha \vee \beta) \equiv (\neg\alpha \vee \neg\beta)$

   $\boxed{c)}$ $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$

4. Step 4: move "or" inwards and "and" outwards

   a) $(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta \vee \gamma)$

   $\boxed{b)}$ $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

   c) $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$

(c) [4 pts] A group of Stanford students write a Convert-to-CNF-ish procedure. In their implementation, they simply apply the *first* equivalence (the one labeled "a)") from each of the four steps in part (b). Show the transformed sentence generated by Convert-to-CNF-ish at each stage, when applied to the input sentence $A \Leftrightarrow (C \vee D)$.

$A \Leftrightarrow (C \vee D)$
$(A \Rightarrow (C \vee D)) \wedge ((C \vee D) \Rightarrow A)$
$(A \vee \neg(C \vee D)) \wedge ((C \vee D) \vee \neg A)$
$(A \vee (\neg C \wedge \neg D)) \wedge ((C \vee D) \vee \neg A)$
$(A \vee \neg C \vee \neg D)) \wedge (C \vee D \vee \neg A)$

(d) [2 pts] Is the final output of the Convert-to-CNF-ish equivalent to the input sentence in part (c)? If not, give a possible world where the input and output sentences have different values.

No.
A counterexample is any model where the two sentences have different truth values. The first clause in the final sentence says $(C \wedge D) \Rightarrow A$ rather than $(C \vee D) \Rightarrow A$. So countereamples are $\{A = false, C = true, D = false\}$ and $\{A = false, C = false, D = true\}$.