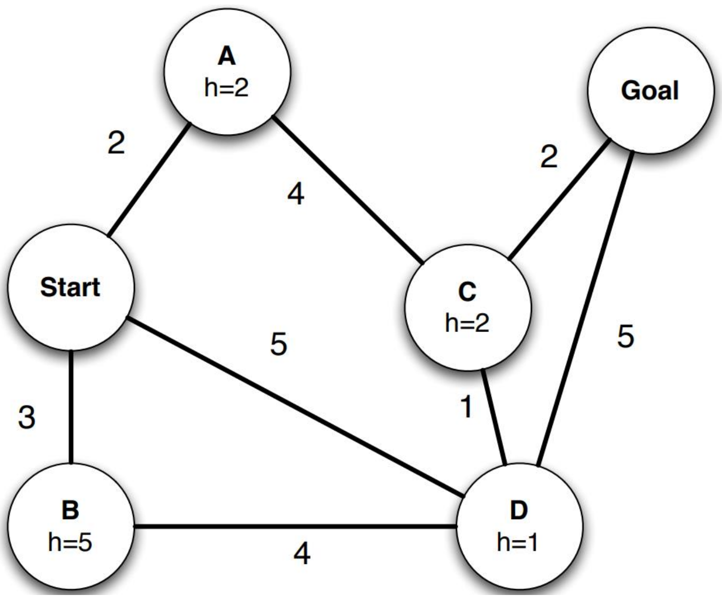
Contents

informed and uninformed search

1) (Graph search algorithms)

For each of the following graph search strategies, work out the order in which states are expanded, as well as the path returned by graph search. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively. Remember that in graph search, a state is expanded only once.



a) Depth-first search.

b) Breadth-first search.

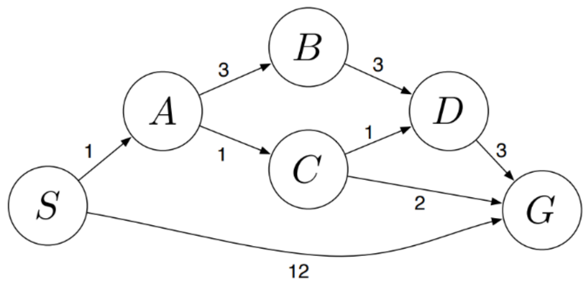
c) Uniform cost search.

d) Greedy search with the heuristic h shown on the graph.

e) A* search with the same heuristic.

2) (Evaluation of heuristics)

Answer the following questions about the search problem shown above. Break any ties alphabetically. For the questions that ask for a path, please give your answers in the form 'S – A – D – G.'



(a) What path would breadth-first graph search return for this search problem?

(b) What path would uniform cost graph search return for this search problem?

(c) What path would depth-first graph search return for this search problem?

(d) What path would A* graph search, using a consistent heuristic, return for this search problem?

(e) Consider the heuristics for this problem shown in the table below.

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

i. Is h1 admissible? Yes No
ii. Is h1 consistent? Yes No
iii. Is h2 admissible? Yes No
iv. Is h2 consistent? Yes No

3) (Status space search)

Pacman and Ms. Pacman are lost in an NxN maze and would like to meet; they don‟t care where. In each time step, both simultaneously move in one of the following directions: {NORTH, SOUTH, EAST, WEST, STOP}. They do not alternate turns. You must devise a plan which positions them together, somewhere, in as few time steps as possible. Passing each other does not count as meeting; they must occupy the same square at the same time.

a. Formally state this problem as a single-agent state-space search problem.

States:

Goal test:

Legal actions (given a state):

Successor function (given a state and an action):

b. Give a non-trivial admissible heuristic for this problem.

c. Circle all of the following graph search methods which are guaranteed to output optimal solutions to this problem:

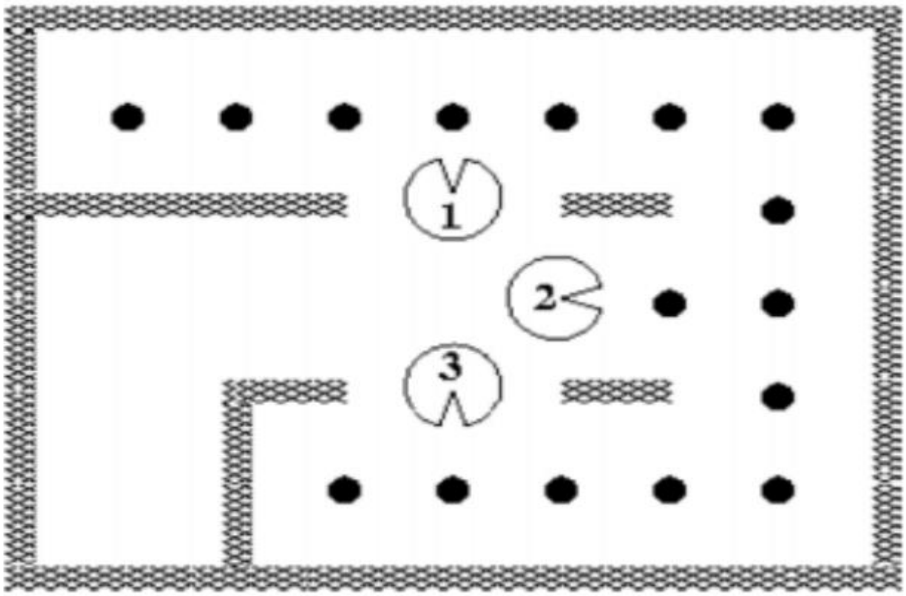
(i) DFS
(ii) BFS
(iii) UCS
(iv) A* (with a consistent and admissible heuristic)
(v) A* (with heuristic that returns zero for each state)

d. If h1 and h2 are admissible, which of the following are also guaranteed to be admissible? Circle all that apply:

(i) h1 + h2
(ii) h1 * h2
(iii) max(h1 , h2)
(iv) min(h1 , h2)
(v) (α)h1+(1−α)h2 for any value α between 0 and 1

4) (Multifactor search)

Pacman is trying eat all the dots, but he now has the help of his family! There are initially k dots, at positions (f1, . . . , fk). There are also n Pac-People, at positions (p1, . . . , pn); initially, all the Pac-People start in the bottom left corner of the maze. Consider a search problem in which all Pac-People move simultaneously; that is, in each step each Pac-Person moves into some adjacent position (N, S, E, or W, no STOP). Note that any number of Pac-People may occupy the same position.



(a) Define the state space of the search problem.

(b) Give a reasonable upper bound on the size of the state space for a general r by c grid.

(c) What is the goal test?

(d) What is the maximum branching factor of the successor function in a general grid?

(e) Circle the admissible heuristics below (−1/2 point for each mistake.)

- $h1(s) = 0$

- $h2(s) = 1$

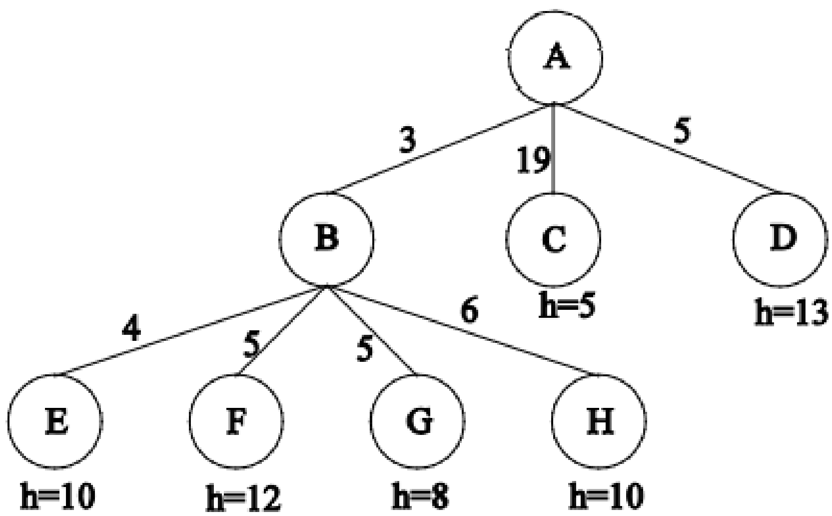
- $h3(s)$ = number of remaining food/ n

- $h4(s) = \max_i \max_j \text{manhattan}(p_i , f_j)$

- $h5(s) = \max_i \min_j \text{manhattan}(p_i , f_j)$

5) (Graph search algorithms)

Consider the following search tree produced after expanding nodes A and B, where each arc is labeled with the cost of the corresponding operator, and the leaves are labeled with the value of a heuristic function, h. For uninformed searches, assume children are expanded left to right. In case of ties, expand in alphabetical order.



Which one node will be expanded next by each of the following search methods?

(a) Depth-First search

(b) Greedy Best-First search

(c) Uniform-Cost search

(d) A* search

6) (search algorithms)

True or False:

(a) Greedy Best-First search using an admissible heuristic is guaranteed to find an optimal solution.

(b) Algorithm A search using the heuristic h(n) = c for some fixed constant c > 0 is guaranteed to find an optimal solution.

(c) If a heuristic is consistent, it is also admissible.

(d) If h1 and h2 are both admissible heuristics, it is always better to use the heuristic h3(n) = max(h1(n), h2(n)) rather than the heuristic h4(n) = min(h1(n), h2(n)).

(e) Beam search with a beam width W = 3 and an admissible heuristic is not guaranteed to find a solution (optimal or not) when one exists.

(f) Say we have a state space where all arc costs are 1 but we don't have a heuristic function. We want to use a space-efficient search algorithm (in terms of the maximum number of nodes stored at any point during the search in Frontier) but also want to guarantee that we find an optimal solution. Which one of the following search methods would be best to use in this situation?

(i) Breadth-First Search
(ii) Depth-First Search
(iii) Iterative-Deepening Search
(iv) Uniform-Cost Search

CSP

1) (Preliminaries and tree design)

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

The classes are:

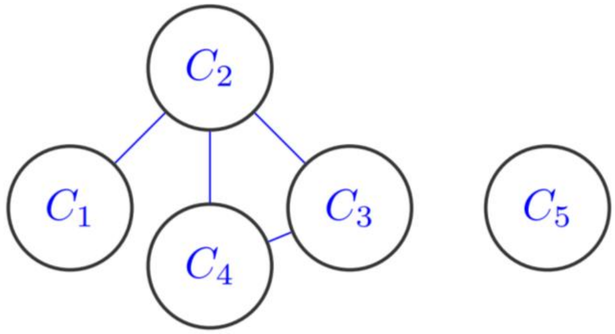
1. Class 1 - Intro to Programming: meets from 8:00-9:00am
2. Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
3. Class 3 - Natural Language Processing: meets from 9:00-10:00am
4. Class 4 - Computer Vision: meets from 9:00-10:00am 5. Class
5 - Machine Learning: meets from 10:30-11:30am

The professors are:
1. Professor A, who is qualified to teach Classes 1, 2, and 5.
2. Professor B, who is qualified to teach Classes 3, 4, and 5.
3. Professor C, who is qualified to teach Classes 1, 3, and 4.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Variables Domains (or unary constraints)

2. Draw the constraint graph associated with your CSP.



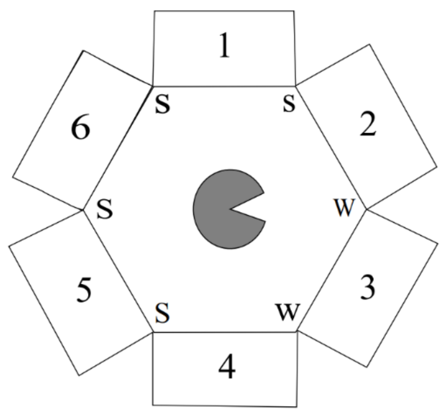
3. Your CSP should look nearly tree-structured. Briefly explain (one sentence or less) why we might prefer to solve tree-structured CSPs.

2) (Circular structure and backtracking)

Pacman is trapped! He is surrounded by mysterious corridors, each of which leads to either a pit (P), a ghost (G), or an exit (E). In order to escape, he needs to figure out which corridors, if any, lead to an exit and freedom, rather than the certain doom of a pit or a ghost.

The one sign of what lies behind the corridors is the wind: a pit produces a strong breeze (S) and an exit produces a weak breeze (W), while a ghost doesn't produce any breeze at all. Unfortunately, Pacman cannot measure the strength of the breeze at a specific corridor. Instead, he can stand between two adjacent corridors and feel the max of the two breezes. For example, if he stands between a pit and an exit he will sense a strong (S) breeze, while if he stands between an exit and a ghost, he will sense a weak (W) breeze. The measurements for all intersections are shown in the figure below.

Also, while the total number of exits might be zero, one, or more, Pacman knows that two



neighboring squares will not both be exits.

Pacman models this problem using variables Xi for each corridor i and domains P, G, and E.

1. State the binary and/or unary constraints for this CSP (either implicitly or explicitly).

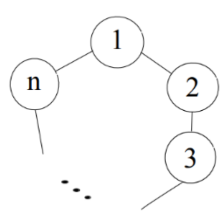
2. Cross out the values from the domains of the variables that will be deleted in enforcing arc consistency.

X_1	P	G	E
X_2	P	G	E
X_3	P	G	E
X_4	P	G	E
X_5	P	G	E
X_6	P	G	E

X_1	P		
X_2		G	E
X_3		G	E
X_4		G	E
X_5	P		
X_6	P	G	E

3. According to MRV, which variable or variables could the solver assign first?

4. Assume that Pacman knows that X6 = G. List all the solutions of this CSP or write none if no solutions exist.



5. The CSP described above has a circular structure with 6 variables. Now consider a CSP forming a circular structure that has n variables (n > 2), as shown below. Also assume that the domain of each variable has cardinality d. Explain precisely how to solve this general class of circle-

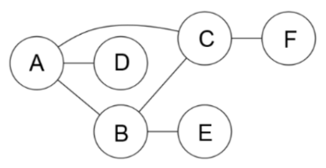
structured CSPs efficiently (i.e. in time linear in the number of variables), using methods covered in class. Your answer should be at most two sentences.

6. If standard backtracking search were run on a circle-structured graph, enforcing arc consistency at every step, what, if anything, can be said about the worst-case backtracking behavior (e.g. number of times the search could backtrack).

3) (Filtering and stability)

(a) The graph below is a constraint graph for a CSP that has only binary constraints. Initially, no variables have been assigned.

For each of the following scenarios, mark all variables for which the specified filtering might result in their domain being changed.



(i) A value is assigned to A. Which domains might be changed as a result of running forward checking for A?

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(ii) A value is assigned to A, and then forward checking is run for A. Then a value is assigned to B. Which domains might be changed as a result of running forward checking for B?

☐A ☐ B ☐ C ☐ D ☐ E ☐ F

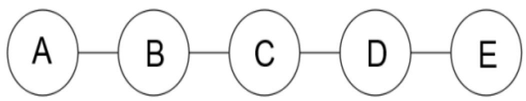
(iii) A value is assigned to A. Which domains might be changed as a result of enforcing arc consistency after this assignment?

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(iv) A value is assigned to A, and then arc consistency is enforced. Then a value is assigned to B. Which domains might be changed as a result of enforcing arc consistency after the assignment to B?

☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

(b) You decide to try a new approach to using arc consistency in which you initially enforce arc consistency, and then enforce arc consistency every time you have assigned an even number of variables. You have to backtrack if, after a value has been assigned to a variable, X, the recursion returns at X without a solution. Concretely, this means that for a single variable with d values remaining, it is possible to backtrack up to d times. For each of the following constraint graphs, if each variable has a domain of size d, how many times would you have to backtrack in the worst case for each of the specified orderings?

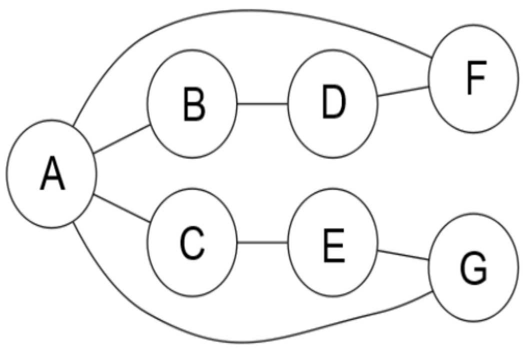


A-B-C-D-E: ____________________

A-E-B-D-C: ____________________

C-B-D-E-A: ____________________

(ii) [6 pts]



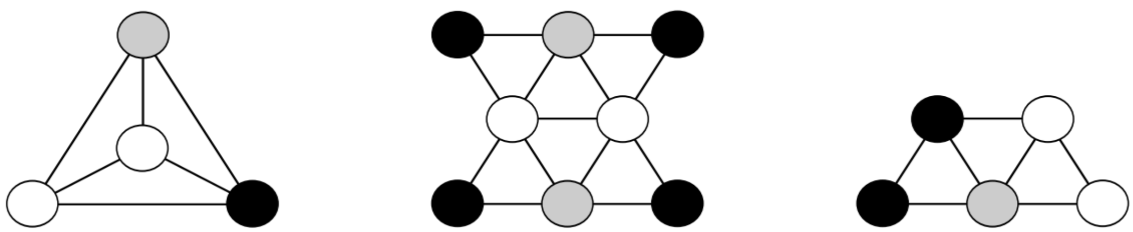
A-B-C-D-E-F-G: ____________________

F-D-B-A-C-G-E: ____________________

C-A-F-E-B-G-D: ____________________

4) (Local search and objective function evaluation)

In this question we are considering CSPs for map coloring. Each region on the map is a variable, and their values are chosen from {black, gray, white}. Adjacent regions cannot have the same color. The figures below show the constraint graphs for three CSPs and an assignment for each one. None of the assignments are solutions as each has a pair of adjacent variables that are white. For both parts of this question, let the score of an assignment be the number of satisfied constraints (so a higher score is better).



(1) (2) (3)

Consider applying Local Search starting from each of the assignments in the figure above. For each successor function, indicate whether each configuration is a local optimum and whether it is a global optimum (note that the CSPs may not have satisfying assignments).

Successor Function	CSP	Local optimum?		Global Optimum?	
Change a single variable	(1)	Yes	No	Yes	No
	(2)	Yes	No	Yes	No
	(3)	Yes	No	Yes	No

	(1)	Yes No	Yes No
Change a single variable, or a pair of variables	(2)	Yes No	Yes No
	(3)	Yes No	Yes No

5) (Processing communication processes)

Four people, A, B, C, and D, are all looking to rent space in an apartment building. There are three floors in the building, 1, 2, and 3 (where 1 is the lowest floor and 3 is the highest). Each person must be assigned to some floor, but it's ok if more than one person is living on a floor. We have the following constraints on assignments:

• A and B must not live together on the same floor.

• If A and C live on the same floor, they must both be living on floor 2.

• If A and C live on different floors, one of them must be living on floor 3.

• D must not live on the same floor as anyone else.

• D must live on a higher floor than C.

We will formulate this as a CSP, where each person has a variable and the variable values are floors.

(a) Draw the edges for the constraint graph representing this problem. Use binary constraints only. You do not need to label the edges.

(b) Suppose we have assigned C = 2. Apply forward checking to the CSP, filling in the boxes next to the values for each variable that are eliminated:

A | 1 ☐ 2 ☐ 3 ☐
B | 1 ☐ 2 ☐ 3 ☐
C | 2 ☐
D | 1 ☐ 2 ☐ 3 ☐

(c) Starting from the original CSP with full domains (i.e. without assigning any variables or doing the forward checking in the previous part), enforce arc consistency for the entire CSP graph, filling in the boxes next to the values that are eliminated for each variable:

A | 1 ☐ 2 ☐ 3 ☐
B | 1 ☐ 2 ☐ 3 ☐
C | 1 ☐ 2 ☐ 3 ☐
D | 1 ☐ 2 ☐ 3 ☐

(d) Suppose that we were running local search with the min-conflicts algorithm for this CSP, and currently have the following variable assignments.

A | 3
B | 1

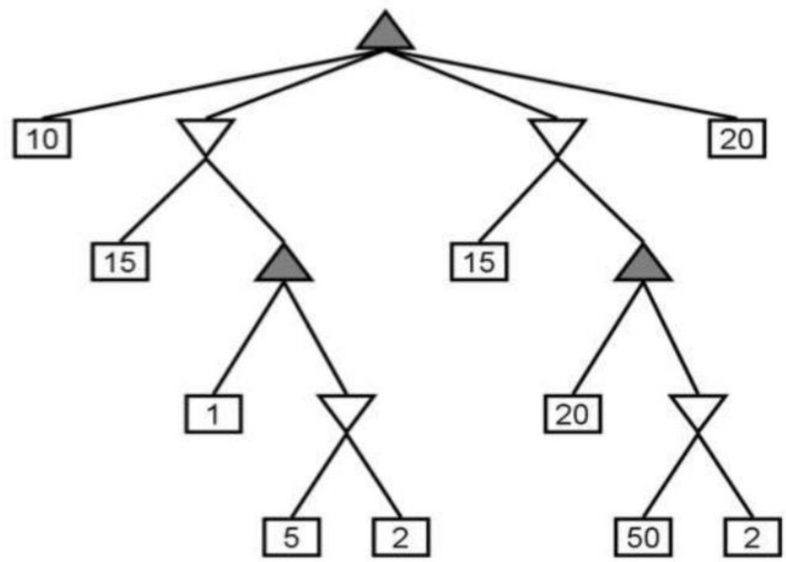
C | 2
D | 3

Which variable would be reassigned, and which value would it be reassigned to? Assume that any ties are broken alphabetically for variables and in numerical order for values.

Adversarial Search

1) (minimax tree and alpha-beta pruning)

Consider the following minimax tree:



a. What is the minimax value for the root?

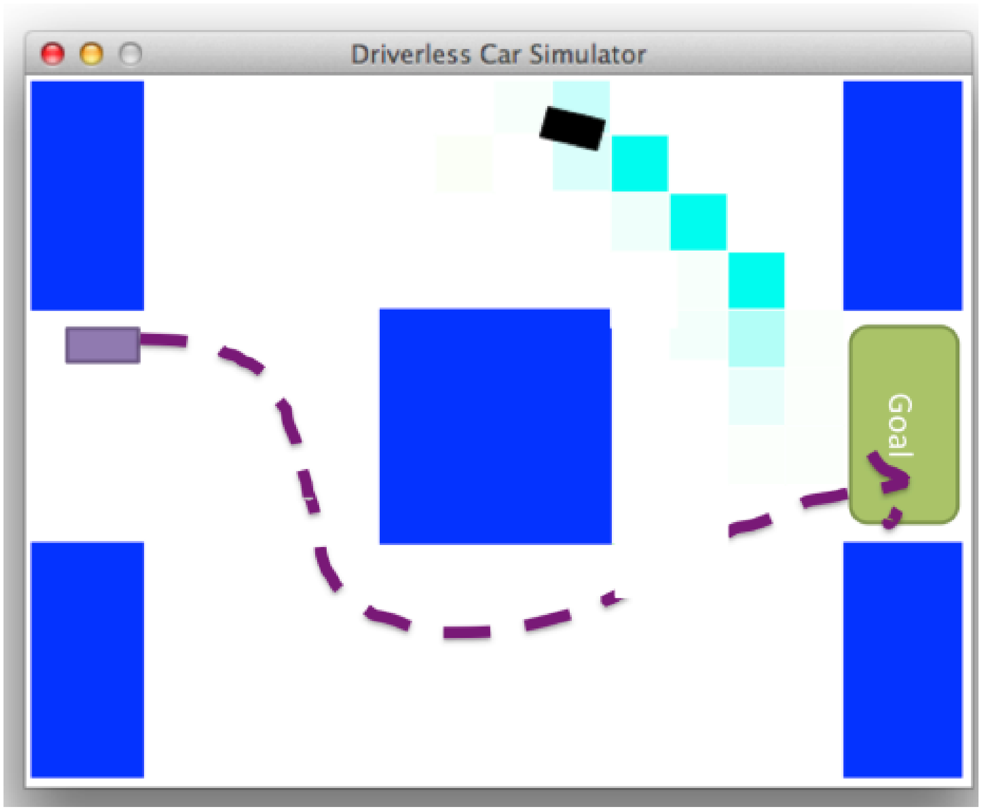
b. Draw an X through any nodes which will not be visited by alpha-beta pruning, assuming children are visited in left-to-right order.

c. Is there another ordering for the children of the root for which more pruning would result? If so, state the order.

d. Propose a general, practical method for ordering children of nodes which will tend to increase the opportunities for pruning. You should be concise, but clearly state both what to do about min nodes and max nodes.

2) (Expectimax and Markov decision(Out of midterm topics))

The goal of this problem is to extend the self driving car to reason about the future and use that reasoning to make a utility maximizing decision on how to act



In this problem we are going to assume that the world is the same as in the Driverless Car programming problem. There is a single agent that exists in a closed world and wants to drive to a goal area. For this problem we are going to assume that there is only one other car.

Each heartbeat (there are twenty per second) each car can perform one of four actions Accelerate, TurnLeft, TurnRight, Brake, None. Assume that for each tile we have a probability distribution that the other car will take each of those actions. A car always has a wheel position and a velocity. Even though the car might not be taking the Accelerate action, if it has positive velocity it may continue to move forward.

[important] For this problem assume that cars are manufactured so precisely that if they take an action there is no uncertainty as to how the car will respond. Also assume that you know the start state of both your car and the other.

Formalize the problem of choosing an action as a Markov decision problem:

(a) What variables make up a state?

(b) What is the start state?

(c) For each state what are the legal actions?

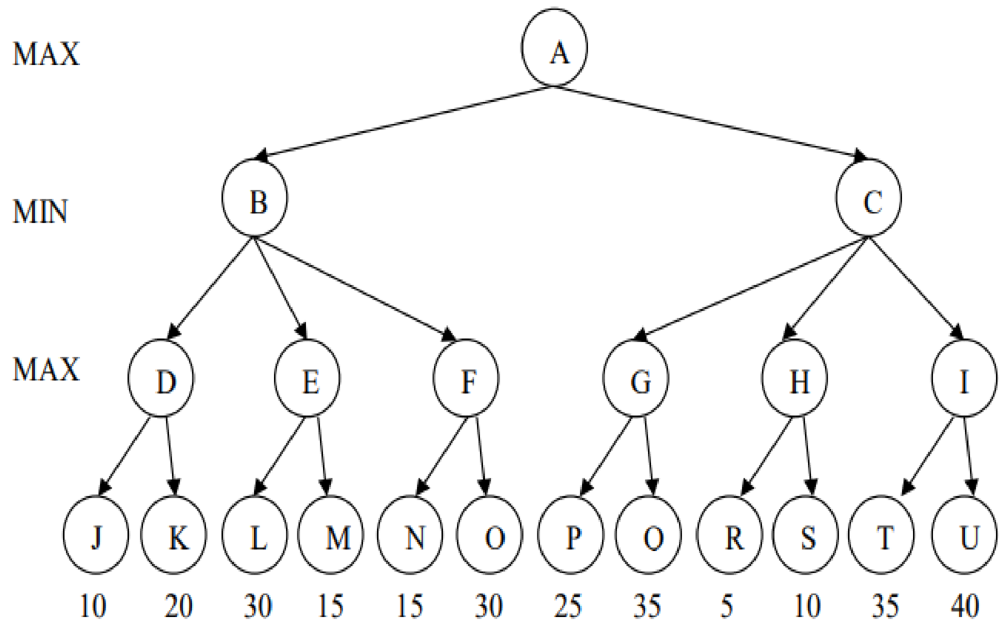
(d) What is a terminal condition for a given state S?

(e) Given a state S from which your agent took action A, what is the successor state distribution?

(f) We are going to solve this problem using expectimax. However, we may not want to expand the entire tree. Give a reasonable heuristic utility for a given state S.

3) (Expectimax and Alpha-Beta Pruning)

the following game tree. The root is a maximizing node, and children are visited left to right.



(a) Ignoring the values given above at the leaf nodes, could there possibly exist some set of values at the leaf nodes in the above tree so that Alpha-Beta pruning would do the following. Answer either Yes or No.

(i) Pruning occurs at the arc from D to K.

(ii) Pruning occurs at the arc from E to M.

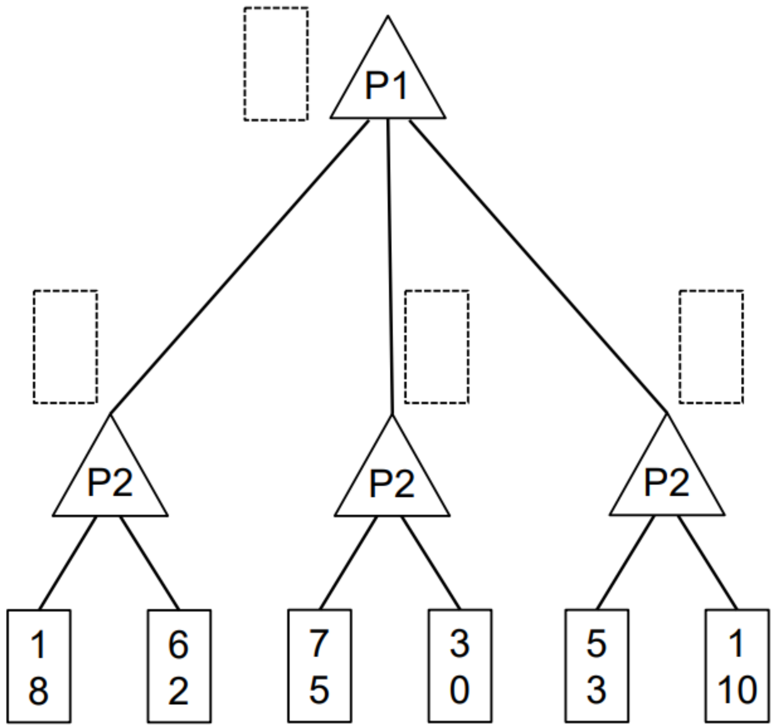
(iii) Pruning occurs at the arc from C to G.

(b) In the tree above assume that the root node is a MAX node, nodes B and C are MIN nodes, and the nodes D, …, I are not MAX nodes but instead are positions where a fair coin is flipped (so going to each child has probability 0.5). What is the Expectimax value at node A?

(c) True or False: When using Expectimax to compute the best move at the root, rescaling the values at all leaf nodes by multiplying them all by 10 may result in a different move at the root.

4) (MiniMax , Max-First Search and Expectimax)

For the following game tree, each player maximizes their respective utility. Let x, y respectively denote the top and bottom values in a node. Player 1 uses the utility function U1(x, y) = x.



(a) Both players know that Player 2 uses the utility function U2(x, y) = x − y.

(i) Fill in the rectangles in the figure above with pair of values returned by each max node.

(ii) You want to save computation time by using pruning in your game tree search. On the game tree above, put an 'X' on branches that do not need to be explored or simply write 'None'. Assume that branches are explored from left to right.

(b) Now assume Player 2 changes their utility function based on their mood. The probabilities of Player 2's utilities and mood are described in the following table. Let M, U respectively denote the mood and utility function of Player 2.

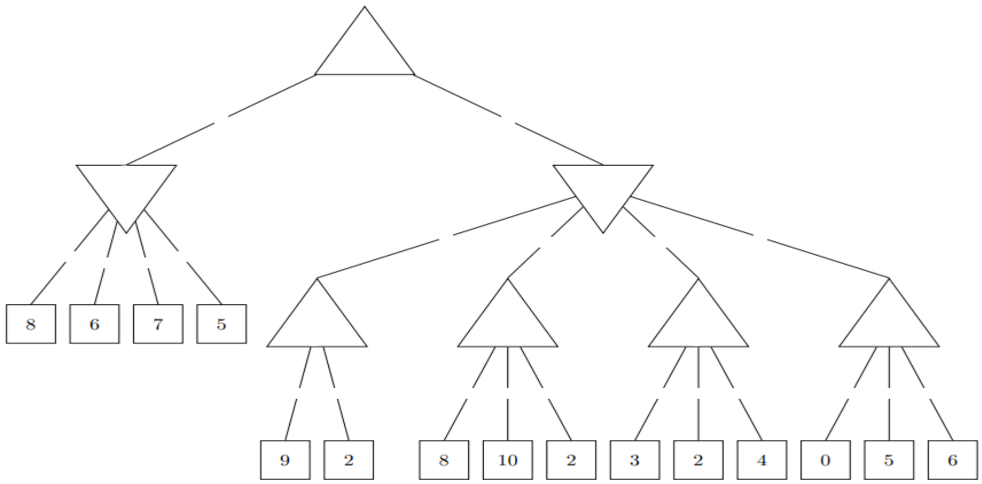
P(M = happy)	P(M = mad)
a	b

	M = happy	M = mad
P(U2(x, y) = −x | M)	c	f
P(U2(x, y) = x − y | M)	d	g
P(U2(x, y) = x 2 + y 2 | M)	e	h

Calculate the maximum expected utility of the game for Player 1 in terms of the values in the game tree and the tables. It may be useful to record and label your intermediate calculations. You may write your answer in terms of a max function.

5) (MiniMax)

(Minimax The first part is based upon the following tree. Upward triangle nodes are maximizer nodes and downward are minimizers. (small squares on edges will be used to mark pruned nodes in part (ii))



(i) Complete the game tree shown above by filling in values on the maximizer and minimizer nodes.

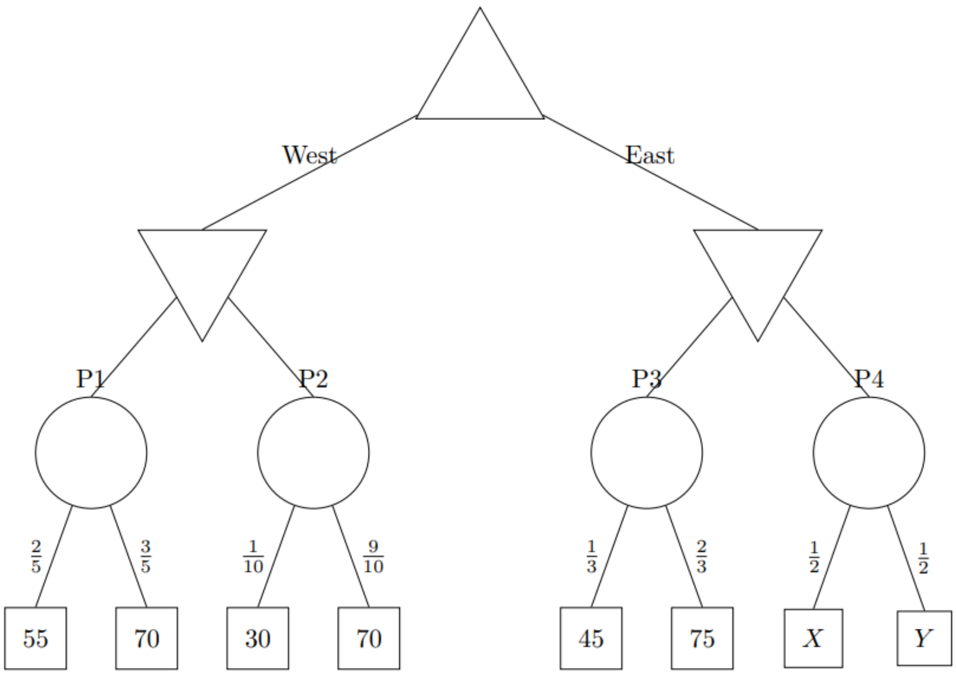
(ii) Can any edges be pruned? Explain.

6) (Game theory and game tree (MiniMax))

Pacman is playing a tricky game. There are 4 portals to food dimensions. But, these portals are guarded by a ghost. Furthermore, neither Pacman nor the ghost know for sure how many pellets are behind each portal, though they know what options and probabilities there are for all but the last portal.

Pacman moves first, either moving West or East. After which, the ghost can block 1 of the portals available.

You have the following gametree. The maximizer node is Pacman. The minimizer nodes are ghosts and the portals are chance nodes with the probabilities indicated on the edges to the food. In the event of a tie, the left action is taken. Assume Pacman and the ghosts play optimally.



(i) Fill in values for the nodes that do not depend on X and Y.

(ii) What conditions must X and Y satisfy for Pacman to move East? What about to definitely reach the P4? Keep in mind that X and Y denote numbers of food pellets and must be whole numbers: X, Y ∈ {0, 1, 2, 3, . . .}.

To move East:

To reach P4: