CS-271P, Intro to A.I. — Mid-term Exam — Winter Quarter, 2018

YOUR NAME: ______________________________

YOUR ID: __________ ID TO RIGHT: __________ ROW: ________ SEAT: ________

Please turn off all cell phones now.

The exam will begin on the next page. Please, do not turn the page until told.

When told to begin, check first to ensure that your copy has all the pages, as numbered 1-15 in the bottom-right corner of each page. We will supply a new exam for any copy problems.

The exam is closed-notes, closed-book. No calculators, cell phones, electronics.

Clear your desk except for pen, pencil, eraser, & water bottle. Put backpacks under your seat. Please do not detach the provided scratch paper from the exam.

After you first stand up from your seat, your exam is over and must be turned in immediately. You may turn in your Midterm exam early and leave class when you are finished.

This page summarizes the points for each question, so you can plan your time.

1. (4 pts total, 1 pt each) TASK ENVIRONMENT.

2. (10 pts total) GENETIC ALGORITHMS: N-QUEENS PROBLEM.

3. (8 pts total, 2 pts each) HYPOTHETICAL ALPHA-BETA PRUNING.

4. (7 pts total, 1 pt each) PROBABILITY FORMULAS.

5. (10 pts total) BAYESIAN NETWORKS.

6. (10 pts total, 2 pts each) STATE-SPACE SEARCH.

7. (12 pts total, 1 pt each) SEARCH QUESTIONS.

8. (10 pts total, 1/2 pt each) SEARCH PROPERTIES.

9. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.

10. (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING.

11. (14 pts total, 1 pt each) AGENT/SEARCH CONCEPTS.

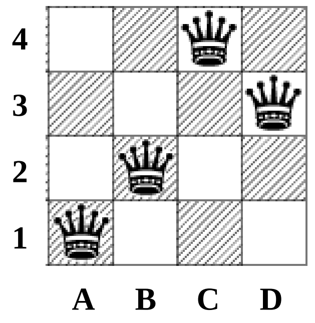
The Exam is printed on both sides to save trees! Work both sides of each page!

1. (4 pts total, 1 pt each) TASK ENVIRONMENT. Your book defines a task environment as a set of four things, with acronym PEAS. Fill in the blanks with the names of the PEAS components. See Section 2.3.1

P<u>erformance (measure)</u> E<u>nvironment</u> A<u>ctuators</u> S<u>ensors</u>

2. (10 pts total. Fractional points will be accumulated across all problems, and then at the end rounded upward in your favor) GENETIC ALGORITHMS: N-QUEENS PROBLEM. You are a robot that is assigned to solve the N-Queens problem, where N = number of rows = number of columns. <u>You choose to use a Genetic Algorithm.</u> Recall that each column contains exactly one queen. A state is represented as a vector of integers wherein vector element *i* gives the row number of the queen in column *i*. Queens attack each other if they are in the same row or diagonal. (Queens also attack each other if they are in the same column, but the representation exploits this constraint by requiring exactly one queen per column.) A solution is a board position in which each queen in each column is assigned to a row and no queen pairs conflict (= no queens attack each other). The value of a state is the number of queen pairs NOT in conflict, so higher values are better.

This problem is about the 4-Queens problem. The example 4-Queens board below gives the representation and value of the state shown. <u>Note that all vector coordinates and row numbers are one-based.</u> See Section 4.1.4



Representation(state shown) = [1 2 4 3]

Value(state shown) = 4

Note: Four queen pairs (A1:C4, A1:D3, B2:C4, B2:D3) are not in conflict

2.a. (4 pts total, 1 pt each) Genetic algorithm concepts for N-queens. For each of the genetic algorithm terms on the left, write in the letter corresponding to the best answer or the correct definition on the right <u>as modified to apply to the N-queens problem.</u> The first one is done for you as an example.

A	Individual	A	A vector of N numbers in which each element <u>*i*</u> is the row number of the queen in column *i* of a given board position
E	Fitness function	B	A set of *k* states, each of which represents a chess board with N queens
D	Mutation	C	The act of combining two chess boards by splitting them at the same random position and exchanging the right-hand sides of each board
C	Crossover	D	The act of moving a randomly chose queen to a new random row on a given chess board
B	Population	E	The number of queens that do not attack each other on a given chess board

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

2.b. (6 pts total, 1/2 pt each. Fractional points will be accumulated across all problems, and then at the end rounded upward in your favor) Genetic algorithm application.

2.b.i. (3 pts total, 1/2 pt each) Genetic algorithm application: Fitness. Consider the initial population shown below. On the line labeled '**Fitness**' indicate the fitness of each individual, as an integer. On the line labeled '**Probability of Selection**' indicate the probability that each individual will be selected for the next generation, as a common fraction. The first one is done for you as an example.

	State1	**State2**	**State3**	**State4**
Individual	[2 4 3 4]	[3 3 1 1]	[2 1 2 4]	[1 1 1 1]
Fitness	3	2	3	0
Probability of Selection	3/8	2/8	3/8	0/8

2.b.ii. (1.5 pts total, 1/2 pt each) Genetic algorithm application: Crossover. State1, State2, and State3 have been selected to contribute to the next generation. Indeed, the highly-fit State1 has been selected twice. State1 and State2 have been paired to become the parents of two children via cross-over, as have State1 and State3. For each pair of **Parents**, a Cross-over point has been selected randomly, as indicated by the '**X**' position within each individual. Fill in the blanks with the two **Children** that will result from each Crossover operation (for a total of four children). The first one is done for you as an example.

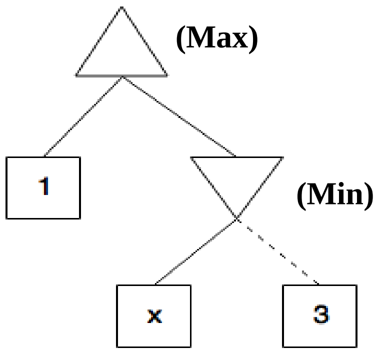
Parents	[2 4 3 **X** 4]	[3 3 1 **X** 1]	*	[2 4 **X** 3 4]	[2 1 **X** 2 4]
Children	[2 4 3 1]	[3 3 1 4]	*	[2 4 2 4]	[2 1 3 4]
Also OK	**[3 3 1 4]**	**[2 4 3 1]**	*	**[2 1 3 4]**	**[2 4 2 4]**

2.b.iii. (1.5 pts total, 1/2 pt each) Genetic algorithm application: Mutation. For each of the following individuals, write the state that would result from mutating that individual at the indicated randomly chosen position to the indicated randomly chosen new value. The first one is done for you as an example.

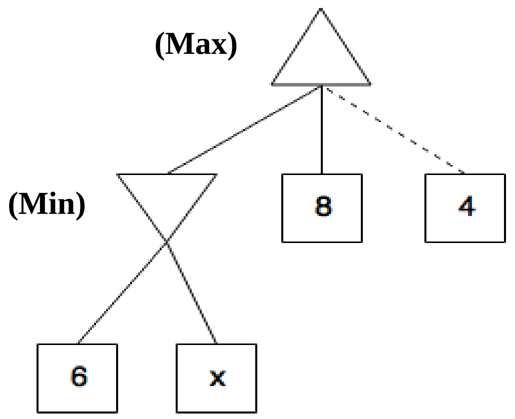
Individual	[3 1 2 3]	[3 3 4 1]	[2 1 2 4]	[1 4 2 3]
Randomly chosen position	3	2	4	1
Randomly chosen new value	4	2	1	3
Resulting mutated vector	[3 1 4 3]	[3 2 4 1]	[2 1 2 1]	[3 4 2 3]

3. (8 pts total, 2 pts each) HYPOTHETICAL ALPHA-BETA PRUNING. For each of the game-trees shown below, indicate for which values of x the dashed branch will be pruned. Write your answer as x ≤ M or x ≥ N, where M and N are integers. If the pruning will not happen for any value of x, write 'None'. If pruning will happen for all values of x, write 'All'. Work each game tree using Depth-First Search in order from left to right. An example tree is given for you below.

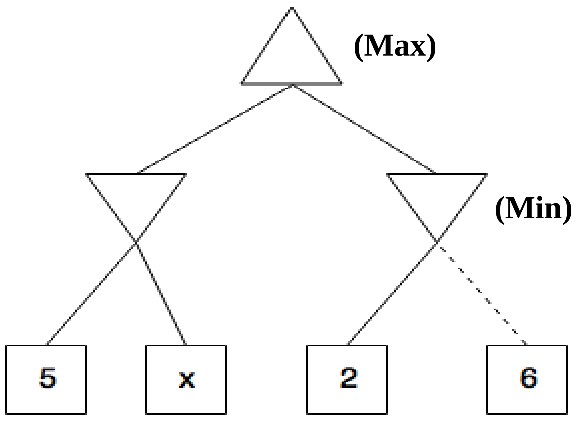
See Section 5.3



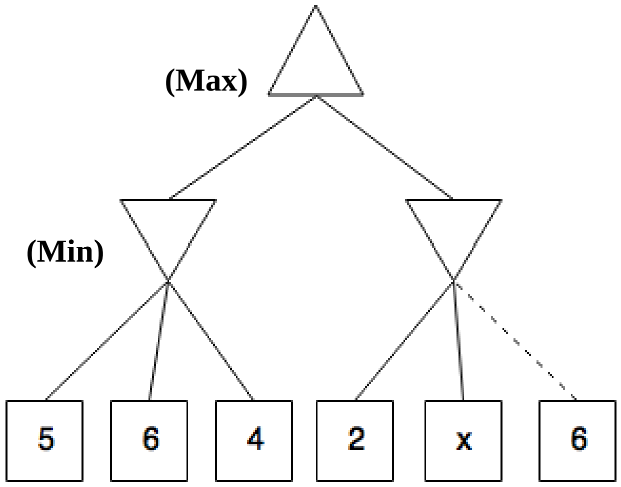
Example: x ≤ 1



3.a. (2 pts) ___None___

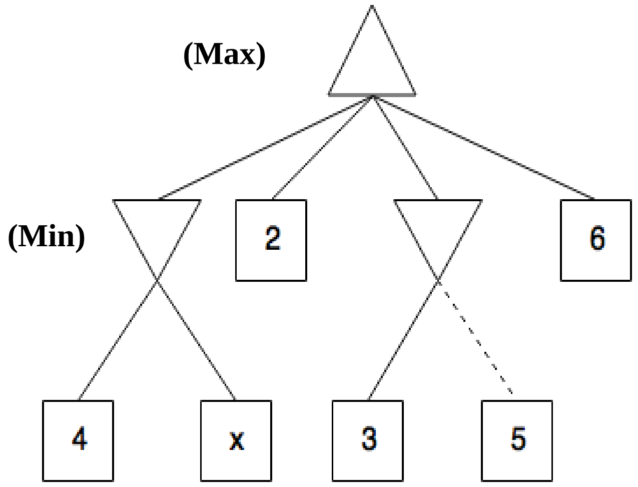


3.b. (2 pts) __x ≥ 2____



3.c. (2 pts) ___All___

Note that both X and the rightmost 6 will be pruned for All values of X.



3.d. (2 pts) ___x ≥ 3___

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

See Chapter 13

4. (7 pts total, 1 pt each) PROBABILITY FORMULAS. Write out the following probability formulas. Below, "in terms of X" means X should appear in your answer. All answers should be formulas, not text.

4.a. (1 pt) Write the formula for $P(A \wedge B)$ in terms of $P(A \vee B)$ and possibly other terms.

Other answers get full credit if they are mathematically correct. E.g., $P(A \wedge B) = P(A \vee B) - P(A \wedge \neg B) - P(B \wedge \neg A)$ is creative, but it gets full credit because it is mathematically correct & responsive to the problem.

$\mathbf{P(A \wedge B)} = P(A) + P(B) - P(A \vee B)$

4.b. (1 pt) Write the formula for the conditional probability $P(A \mid B)$.

$\mathbf{P(A \mid B)} = P(A \wedge B) / P(B)$

4.c. (1 pt) Factor $P(A \wedge B \wedge C)$ completely using the Product Rule (or Chain Rule). You may use any variable ordering you wish.

$\mathbf{P(A \wedge B \wedge C)} = P(A \mid B \wedge C) * P(B \mid C) * P(C)$

Other variable orderings are OK iff correct, e.g., $P(A \wedge B \wedge C) = P(C \mid A \wedge B) * P(B \mid A) * P(A)$ $= P(B \mid A \wedge C) * P(C \mid A) * P(A)$, etc.

4.d. (1 pt) Given a joint probability distribution $P(A \wedge B \wedge C)$, use the Sum Rule (or Law of Total Probability) to write the marginal probability of $P(A)$.

$\mathbf{P(A)} = \Sigma_{B,C}\, P(A \wedge B \wedge C)$

All are correct:

$P(A) = \Sigma_B \Sigma_C\, P(A \wedge B \wedge C)$

$= \Sigma_{b \in B} \Sigma_{c \in C}\, P(A \wedge b \wedge c)$

$= \Sigma_{b \in B,\, c \in C}\, P(A \wedge b \wedge c)$

4.e. (1 pt) Write Bayes' Rule (or Bayes' Theorem).

$$\mathbf{P(A \mid B)} = P(B \mid A) * P(A) / P(B) = \frac{P(B \mid A) * P(A)}{\Sigma_{a \in A}\, P(B \mid a) * P(a)} = \frac{P(B \mid A) * P(A)}{P(B \mid a) * P(a) + P(B \mid \neg a) * P(\neg a)}$$

Iff A is Boolean.

4.f. (1 pt) Assume that A and B of P(A) and P(B) and possibly other terms.

Bayes' Rule is written in several different forms in different places, any of which gets full credit if mathematically correct.

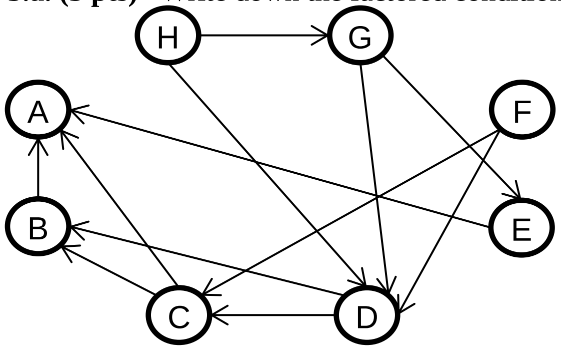
$\mathbf{P(A \wedge B)} = P(A) * P(B)$

4.g. (1 pt) Assume that A and B are conditionally independent given C. Write $P(A \wedge B \mid C)$ in terms of $P(A \mid C)$ and $P(B \mid C)$ and possibly other terms.

$\mathbf{P(A \wedge B \mid C)} = P(A \mid C) * P(B \mid C)$

5. (10 pts total) BAYESIAN NETWORKS.

5.a. (3 pts) Write down the factored conditional probability expression corresponding to this Bayesian Network:

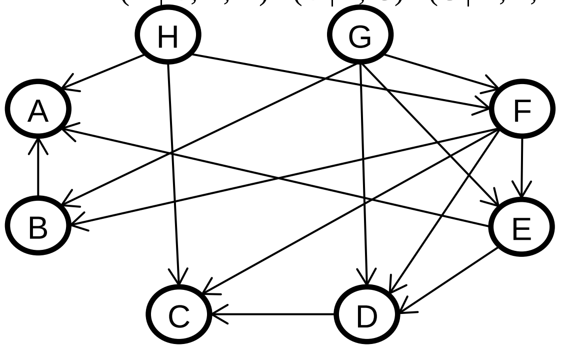


See Chapter 14

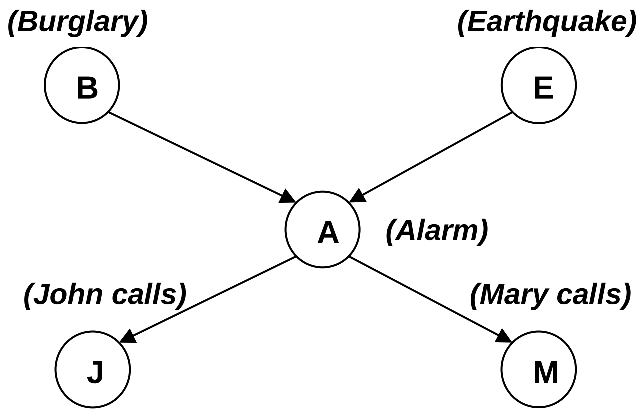
P(A | B, C, E) P(B | C, D) P(C | D, F) P(D | F, G, H) P(E | G) P(F) P(G | H) P(H)

5.b. (3 pts) Draw the Bayesian Network corresponding to this factored conditional probability expression:

P(A | B, E, H) P(B | F, G) P(C | D, F, H) P(D | E, F, G) P(E | F, G) P(F | G, H) P(G) P(H)



5.c. (4 pts) Shown below is the Bayesian network corresponding to the Burglar Alarm problem, i.e., P(J,M,A,B,E) = P(J | A) P(M | A) P(A | B, E) P(B) P(E). This is Fig. 14.2 in your R&N textbook.



P(B)
.001

P(E)
.002

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01

Write down an expression that will evaluate to P(J=t $\wedge$ M=f $\wedge$ A=t $\wedge$ B=f $\wedge$ E=t). **Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols.** You do not need to carry out the multiplication to produce a single number (probability). **SHOW YOUR WORK.**

P(J=t $\wedge$ M=f $\wedge$ A=t $\wedge$ B=f $\wedge$ E=t)

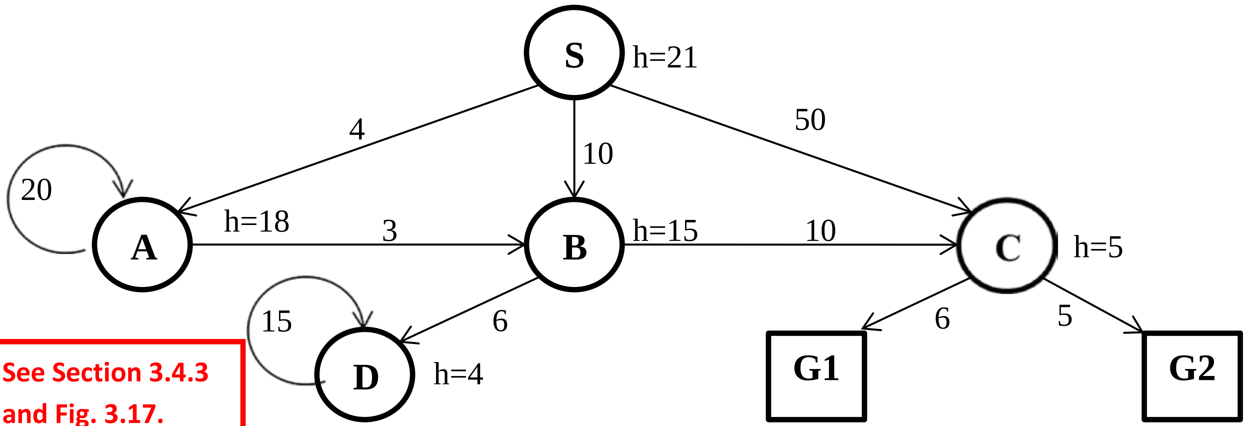
= P(J=t | A=t) * P(M=f | A=t) * P(A=t | B=f $\wedge$ E=t) * P(B=f) * P(E=t)

= .90 * .30 * .29 * .999 * .002

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

6. (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. **(Note: A, D are successors of themselves.)** Successor nodes are returned in left-to-right order. (The successor nodes of S are A, B, C; the successor nodes of A are A, B; the successor nodes of B are D, C; and the successor nodes of C are G1, G2. For LIFO and FIFO queues the children will be processed in those node orders, i.e., assume that the child list is concatenated to the front or back of the queue in the order stated above. Priority queues are always sorted by the queue sort function.)

The start node is S and there are two goal nodes, G1 and G2. For each search strategy below, indicate **(1) the order** in which nodes are expanded, and **(2) the path and cost** to the goal that was found, if any. Write "None" for the path and cost if the goal was not found. The first one is done for you, as an example.



See Section 3.4.3 and Fig. 3.17.

6.a. (example) DEPTH-FIRST SEARCH:

6.a.i Order of expansion: S A A A A A A ...

6.a.ii Path to goal found: None Cost of path found: None

6.b. (2 pts) BREADTH-FIRST SEARCH:

See Section 3.4.1 and Fig. 3.11.

on: S A B C (G1)

6.b.ii Path to goal found: S C G1 Cost of path found: 56

6.c. (2 pts) ITERATIVE DEEPENING SEARCH:

See Sections 3.4.4-5 and Figs. 3.18-19.

S S A B C (G1)

6.c.ii Path to goal found: S C G1 Cost of path found: 56

6.d. (2 pts) UNIFORM COST SEARCH:

See Section 3.4.2 and Fig. 3.14.

on: S A B D C (G2) or S A B B D D C C (G2)

nd: S A B C G2 Cost of path found: 22

6.e. (2 pts) GREEDY BEST FIRST SEARCH:

See Section 3.5.1 and Fig. 3.23.

on: S C (G1) or S C (G2)

d: S C G1 or S C G2 Cost of path found: 56 or 55

6.f. (2 pts) A* SEARCH:

See Section 3.5.2 and Figs. 3.24-25.

n: S A B D C G2

d: S A B C G2 Cost of path found: 22

7. (12 pts total, 1 pt each) SEARCH QUESTIONS. Label the following as T (= True) or F (= False).

7.a. (1 pt) ____T______ An admissible heuristic NEVER OVER-ESTIMATES the remaining cost (or distance) to the goal.

7.b. (1 pt) ____F______ Greedy search is both complete and optimal when the heuristic is optimal.

7.c. (1 pt) ____T______ A consistent heuristic NEVER VIOLATES the triangle inequality.

For 7.e and 7.k: The branching factor b in state space search is always finite because state space search is impossible with an infinite branching factor. If b were infinite for any state s, then Expand(s) would never return, and so the entire state space process would fail. Note that standard graph search algorithms also prohibit an infinite branching factor, e.g., Dijkstra's algorithm cannot even read in any graph that contains a node with infinitely many arcs.

7.d. (1 pt) ____F______ If the searc
the only local maximum), then hill
maximum.

7.e. (1 pt) ____T______ A* search i
cost is bounded away from zero by

7.f. (1 pt) ____F______ Hill-climbing has very attractive space properties because it uses only *O*(*bd*) space.

7.g. (1 pt) ____F______ Simulated annealing will accept more and worse bad moves at a low temperature than at a high temperature.

7.h. (1 pt) ____T______ Simulated
optima.

For 7.k: Let C* be the true cost to the optimal goal, epsilon the minimum step cost, and b the maximum branching factor. Then there are at most floor(C*/epsilon) steps from the root to the optimal goal. At most b^[1 + floor(C*/epsilon)] nodes will be expanded before the optimal goal is found. The optimal goal will sort in front of any and all suboptimal goals. Thus, UCS is both complete and optimal.

7.i. (1 pt) ____F______ The simula

7.j. (1 pt) ____T______ If the searc
leaves (e.g., in tic-tac-toe), then it

7.k. (1 pt) ____T______ Uniform-cost search is both complete and optimal when the minimum step cost is bounded away from zero by a small positive constant.

7.l. (1 pt) ____F______ Mini-Max search assumes that the opponent plays optimally, so the opponent can defeat it by playing sub-optimally.

8. (10 pts total, 1/2 pt each) SEARCH PROPERTIES. Fill in the values of the four evaluation criteria for each search strategy. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; C* is the cost of the optimal solution; step costs are identical and equal to some positive ε; and in Bidirectional search both directions use breadth-first search. Note that these conditions satisfy all of the footnotes of Fig. 3.21 in your book. **See Figure 3.21**

Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First	Yes	O(b^d)	O(b^d)	Yes
Uniform-Cost	Yes	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	Yes
Depth-First	No	O(b^m)	O(bm)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
Bidirectional (if applicable)	Yes	O(b^(d/2))	O(b^(d/2))	Yes

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

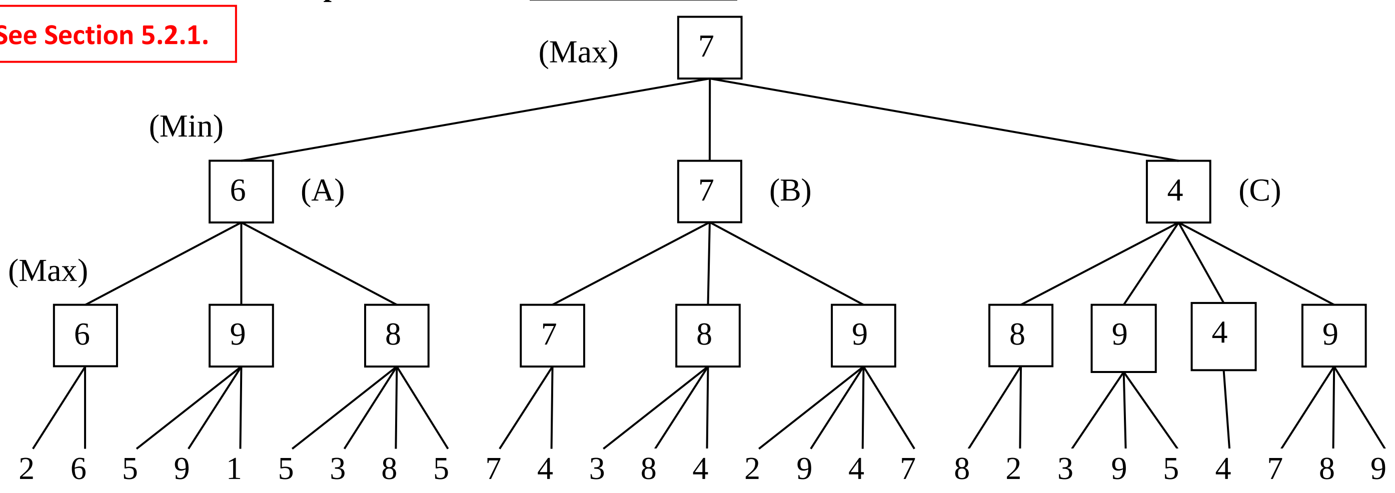
9. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES. The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

9.a. Fill in each blank square with the proper mini-max search value.

9.b. What is the best move for Max? (write A, B, or C) B

9.c. What score does Max expect to achieve? 7

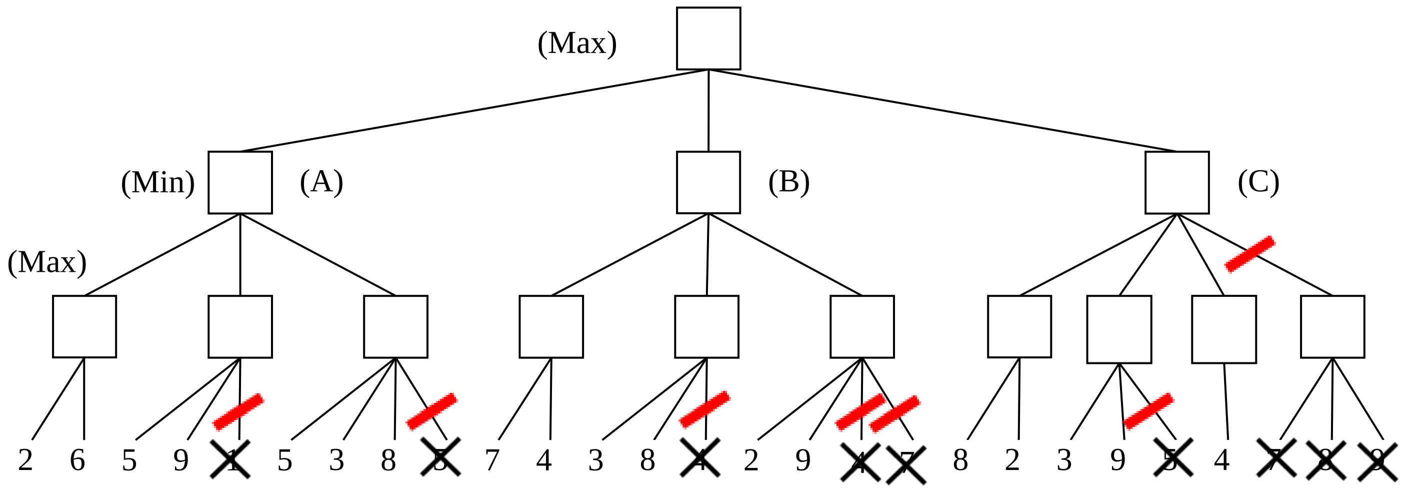
See Section 5.2.1.



10. (10 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (1.a). You do not need to indicate the branch node values again.

Cross out each leaf node that will be pruned by Alpha-Beta Pruning.

See Section 5.3.



11. (14 pts total, 1 pt each) AGENT/SEARCH CONCEPTS. For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

A	Agent	A	Perceives environment by sensors, acts by actuators
K	Percept	B	All states reachable from the initial state by a sequence of actions
L	Rational Agent	C	Guaranteed to find a solution if one is accessible
B	State Space	D	Process of removing detail from a representation
I	Search Node	E	Maximum number of successors of any node
N	Link between nodes	F	Set of all leaf nodes available for expansion at any given time
J	Path	G	Estimates cost of cheapest path from current state to goal state
D	Abstraction	H	Guaranteed to find lowest cost among all accessible solutions
H	Optimal Search	I	Represents a state in the state space
C	Complete Search	J	Sequence of states connected by a sequence of actions
M	Expand a state	K	Agent's perceptual inputs at any given instant
F	Frontier	L	Agent that acts to maximize its expected performance measure
O	Search Strategy	M	Apply each legal action to a state, generating a new set of states
E	Branching Factor	N	Represents an action in the state space
G	Heuristic Function	O	How a search algorithm chooses which node to expand next

****** THIS IS THE END OF THE MID-TERM EXAM ******