CS-171, Intro to A.I. — Final Exam — Fall Quarter, 2018

NAME:______________________________________ UCI NetID: ____________________

YOUR ID#: _________ ID# TO RIGHT: _________ ID# TO LEFT: ___________ 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 1-14 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. (12 pts total, 3 pts each) FOPL Unification/Resolution.
2. (10 pts total, -2 for each error, but not negative) Game Search.
3. (12 pts total, 3 pts each) Probability formulas.
4. (10 pts total, 2 pts each) English to FOL Conversion.
5. (12 pts total, 1 pt each) State Space Search.
6. (10 pts total, -1 for each mistake, but not negative) Bayesian Networks.
~~**7. (12 pts total, -2 for each mistake, but not negative) Local Search.**~~
8. (12 pts total, 1 pt each) SEARCH QUESTIONS.
9. (10 pts total, 1/2 pt each) SEARCH PROPERTIES.

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

1. (12 pts total, 3 pts each) FOPL Unification/Resolution. Use your knowledge of resolution and unification to unify and resolve the given FOPL clauses. For each pair of FOPL clauses below:

(1) Write the most general unifier (or MGU) that will allow you to perform resolution on those clauses, or write "None" if none is possible. Write your answer in the form of a substitution as given in your book, e.g. the substitution {x/John} means substitute x by John.

(2) Perform resolution on the two clauses and write the resulting clause, or write "True" if the resulting clause simplifies to True.

The first one is done for you as an example.

1.example.

¬ Rich(x) ∨ Unhappy(x)
Rich(John) **with θ =** {x/John}

Unhappy(John)

1.a. (2 pts total, 1 pt each)

Knows(John, x)
¬ Knows(y, Elizabeth) ∨ Likes(y, x) **with θ =** {x/Elizabeth, y/John}

Likes(John, Elizabeth)

1.b. (2 pts total, 1 pt each)

¬Water(x) ∨ ¬Food(x) ∨ InDesert(x, cameraman(x))
Food(BearGrylls) **with θ =** {x/BearGrylls}

¬Water(BearGrylls) ∨ InDesert(BearGrylls, cameraman(BearGrylls))

1.c. (2 pts total, 1 pt each)

Actor(x) ∨ ¬Tall(x)
¬Actor(TonyStark) ∨ Tall(TonyStark) ∨ Movie(IronMan) **with θ =** {x/TonyStark}

True

1.d. (2 pts total, 1 pt each)

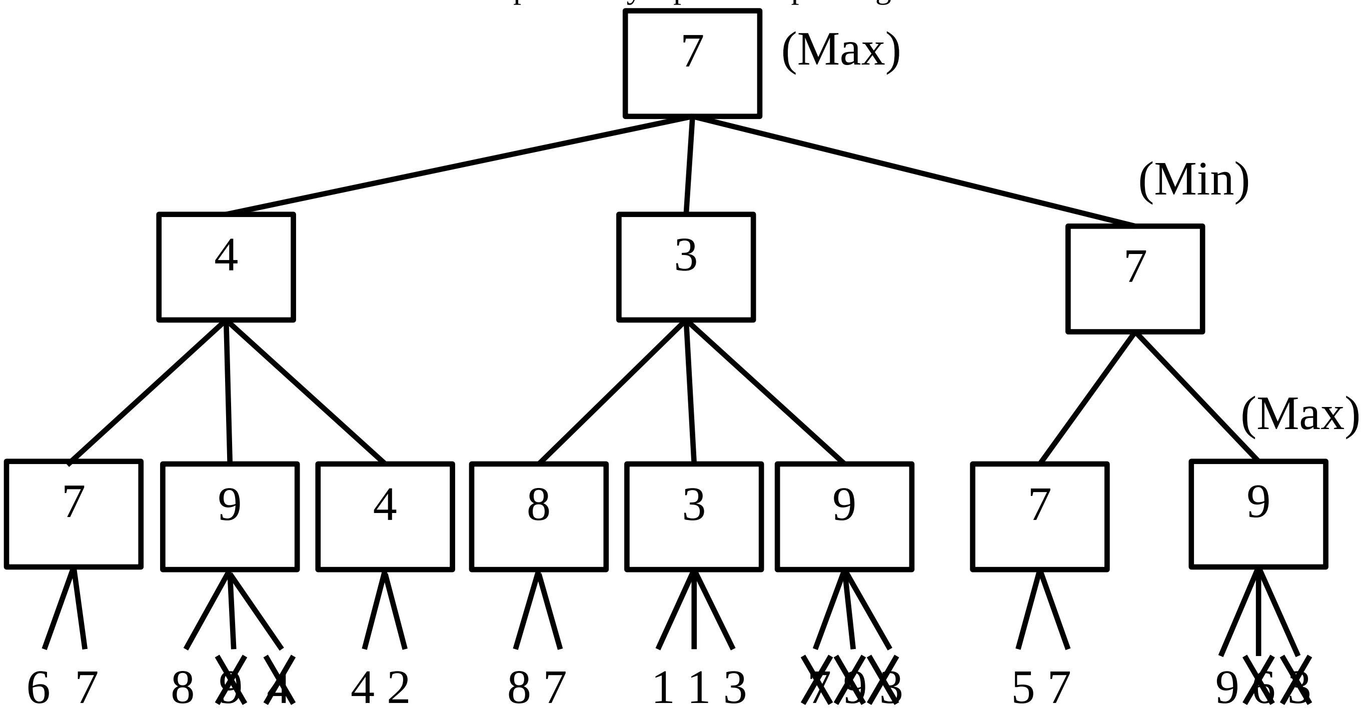
Neighbor(Penny, Friend(x)) ∨ Knows(x, y)
¬ Neighbor(y, Friend(Brother(Sheldon))) **with θ =** {x/Brother(Sheldon), y/Penny}

Knows(Brother(Sheldon), Penny)

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

2. (10 pts total, -2 for each error, but not negative) Game Search. In the game tree below, it is Max's turn to move. Inside each leaf node is the estimated score of that resulting position as returned by the heuristic.

2.a. Perform Mini-Max search. Fill in each box with the value obtained from the mini-max search.
2.b. Cross out each node that would be pruned by alpha-beta pruning.



3. (12 pts total, 3 pts each) Probability formulas. For each term on the left, write the letter of the best description on the right. **The first one is done for you as an example.**

A	Probability Theory	A	Assigns each sentence a degree of belief ranging from 0 to 1
C	Conditional independence	B	$P(a \land b) = P(a)\, P(b)$
B	Independence	C	$P(a \land b \mid c) = P(a \mid c)\, P(b \mid c)$
E	Product rule (chain rule)	D	$P(a \mid b) = P(b \mid a)\, P(a) / P(b)$
D	Bayes' rule	E	$P(a \land b \land c) = P(a \mid b \land c)\, P(b \mid c)\, P(c)$

4. (10 pts total, 2 pts each) English to FOL Conversion. For each English sentence below, write the FOL sentence that best expresses its intended meaning.

Use **Student(x)** to mean that x is a student.
Use **Semester(y)** to mean that y is a semester/year.
Use **Took(s, c, y)** to mean that student s took course c in semester/year y.
Use **Passed(s, c, y)** to mean that student s passed course c in semester/year y.

The first one is done for you as an example.

4.example. Some student took French in Fall2018.

$\exists x \; Student(x) \land Took(x, French, Fall2018)$

4.a. Some student took French in Fall2018 and did not pass.

$\exists x \; Student(x) \land Took(x, French, Fall2018) \land \neg Passed(x, French, Fall2018)$

4.b. Every student in every semester who took French passed it.

$\forall x \; \forall y \; Student(x) \land Semester(y) \land Took(x, French, y) \Rightarrow Passed(x, French, y)$

4.c. There is some semester in which every student who took French passed it.

$\exists y \; \forall x \; Semester(y) \land [\; Student(x) \land Took(x, French, y) \Rightarrow Passed(x, French, y) \;]$

4.d. There is some semester in which some student took French and did not pass it.

$\exists x \; \exists y \; Student(x) \land Semester(y) \land Took(x, French, y) \land \neg Passed(x, French, y)$

4.e. Every semester there is some student who took French and passed it.

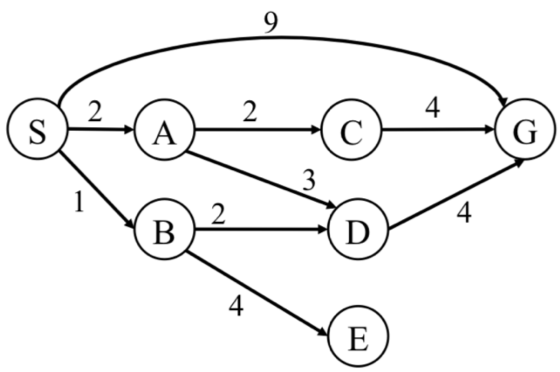
$\forall y \; \exists x \; Semester(y) \Rightarrow Student(x) \land Took(x, French, y) \land Passed(x, French, y)$

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

5. (12 pts total, 1 pt each) State Space Search. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). Step costs are given next to each arc, and heuristic values are given in the Heuristic table below. The successors of each node are indicated by the arrows out of that node. Successors are returned in top-to-bottom order (successors of S are G, A, B; successors of A are C and D; and successors of B are D and E; in that order). The start node is S and the goal node is G.

For each search strategy below, indicate (1) the order in which nodes are expanded; (2) the path to the goal that was found, if any; and (3) the cost of the path. Write "None" for path/cost if the goal was not found.

The first one is done for you as an example.



Heuristic

S	A	B	C	D	E	G
6	0	6	4	1	10	0

5.example. DEPTH-FIRST SEARCH:

5. example.(1) Order of expansion: _S G____

5. example.(2) Path to goal found: _S G_________________________ Cost of path to goal: __9____

5.a. (3 pts total, 1 pt each) BREADTH-FIRST SEARCH:

5.a.(1) Order of expansion: _S G____

5.a.(2) Path to goal found: _S G_________________________ Cost of path to goal: __9____

5.b. (3 pts total, 1 pt each) UNIFORM COST SEARCH:

5.b.(1) Order of expansion: _S B A D C D E G_______________________________________

5.b.(2) Path to goal found: _S B D G______________________ Cost of path to goal: __7____

5.c. (3 pts total, 1 pt each) GREEDY BEST FIRST SEARCH:

5.c.(1) Order of expansion: _S G____

5.c.(2) Path to goal found: _S G_________________________ Cost of path to goal: _9_____

5.d. (3 pts total, 1 pt each) A* SEARCH:

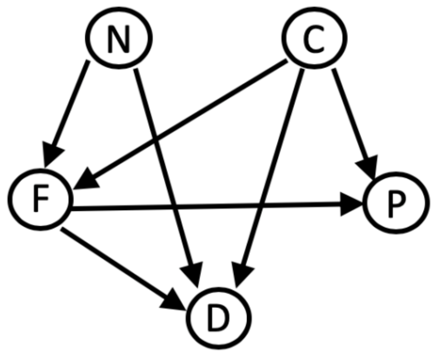
5.d.(1) Order of expansion: _S A D B D G____

5.d.(2) Path to goal found: _S B D G______________________ Cost of path to goal: _7_____

6. (10 pts total, -1 for each mistake, but not negative) Bayesian Networks. This question asks about a medical patient and conditions of nausea (N), cough (C), fever (F), pneumonia (P), and visiting a doctor (D).

- Nausea (N) and Cough (C) influence whether or not the patient may have a Fever (F).
- Cough (C) and Fever (F) influence whether or not the patient may have Pneumonia (P).
- Nausea (N), Cough (C), and Fever (F) influence whether the patient goes to see the doctor (D).
- In this question, assume that the probability of Nausea (N) and Cough (C) are independent.

6.1. (5 pts total, -1 for each mistake, but not negative) Draw the Bayesian Network corresponding to the probabilistic assumptions above.

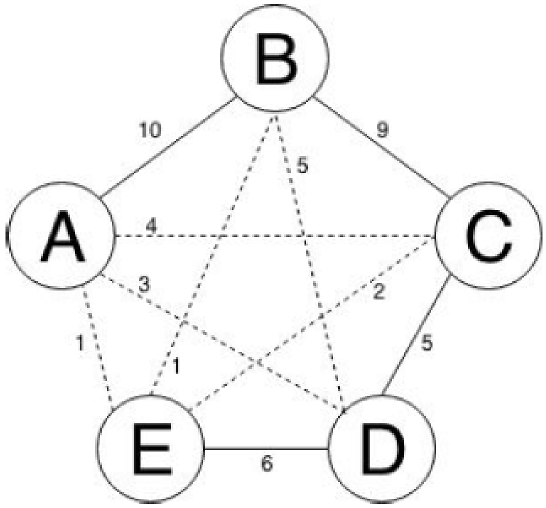


6.2. (5 pts total, -1 for each mistake, but not negative) Write down the corresponding factored form of the joint probability distribution. (Your answer here will be considered correct if it is correct relative to your Bayesian Network in problem 6.1 above, even if your answer to 6.1 was wrong.)

P(N, C, F, P, D) = ___________P(N) P(C) P(F|N, C) P(D|N, C, F) P(P|C, F)______________

7. (12 pts total, -2 for each mistake, but not negative) Local Search. Consider this graph. The Traveling Salesman Problem (TSP) is to construct a path through the graph where (1) each node must be visited once, and (2) no node can be visited twice. Additionally, you seek the lowest-cost path satisfying those constraints.

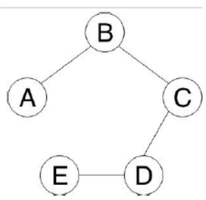
Question #7 was canceled. Everyone gets it question right, regardless of what they answered. After review of student feedback and discussion among the CS 171 Teaching Staff, we agreed that the question was poorly worded and did not capture properly our intended meaning.



You are a robot assigned to find a low-cost Traveling Salesman path through the graph above. You choose to use a local search. You will be given an initial state (a path) that satisfies the Traveling Salesman constraints, but is high-cost. A neighbor (successor) state is constructed by selecting an edge, deleting it, and replacing it with a new edge that maintains the Traveling Salesman constraints on the new state. Your goal is to move from neighbor to neighbor, always maintaining the Traveling Salesman constraint, seeking lower-cost paths.

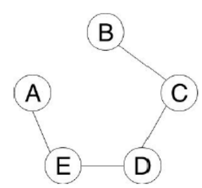
For a variable selection heuristic, select the edge with the current highest weight. Replace it with the edge of lowest weight that satisfies the constraints of the system. Write the total cost of the state next to each state. Continue until the algorithm has completed.

Initial State:



Cost = 30

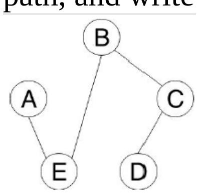
The first one is done for you as an example:



Cost = 21

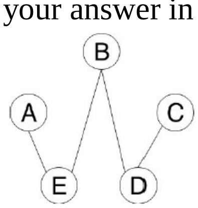
Continue the local search. At each step, select highest-cost edge and replace it with the edge of lowest weight that satisfies the Traveling Salesman constraints. Write the total cost of each state next to that state.

7.a. (6 pts total, -2 for each mistake, but not negative) Draw the next lowest-cost neighbor of the above-right path, and write its cost in the blank provided.



Cost = ______16______

7.b. (6 pts total, -2 for each mistake, but not negative) Draw the next lowest-cost neighbor of the above path in 7.a, and write its cost in the blank provided. (Your answer will be considered correct if it is correct relative to your answer in 7.a above, even if your answer to 7.a was wrong.)



Cost = ______12______

8. (12 pts total, 1 pt each) SEARCH QUESTIONS. Label the following as T (= True) or F (= False).

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

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

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

8.d. (1 pt) ____F______ If the searc[h] ... the only local maximum), then hill ... maximum.

8.e. (1 pt) ____T______ A* search i[s] ... cost is bounded away from zero by ...

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.

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

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

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

8.i. (1 pt) ____F______ The simula...

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

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.

8.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.

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

9. (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

****** THIS IS THE END OF THE FINAL EXAM ******