CS 4700: Foundations of Artificial Intelligence
Spring 2020
Prelim Prep Questions

1. True/False Questions:
    a. If the third state visited by both depth-first and breadth-first search is the same (starting from the same initial state with the same goal and using the same tie-breaking strategy) then they must have visited the same second state.
    b. If you don't check for cycles then breadth-first search can fail to find a solution.
    c. If iterative deepening finds a solution for a given problem then depth-first search will also find that solution.
    d. If breadth-first search is able to find a solution to a search problem, then A* is guaranteed to find a solution.
    e. If h_1, h_2, and h_3 are all admissible functions, then $\frac{h_1}{6} + \frac{h_2}{3} + \frac{h_3}{2}$ is admissible.
    f. Alpha-beta pruning will never result in a different move being selected at the root of the game tree compared to plain minimax search, but the value it generates for the root may differ from that of plain minimax search.
    g. If a sentence in propositional logic is unsatisfiable then its negation is a tautology.
    h. If α and β are clauses in propositional logic, and all the literals in α are contained in β, then $\alpha \models \beta$.
    i. For any propositional sentences α, β, and γ, if $\alpha \models (\beta \land \gamma)$ then $\alpha \models \beta$ and $\alpha \models \gamma$.
    j. If you know $(P \lor Q \lor R)$ and $(\neg P \lor \neg Q \lor R)$, resolution allows you to conclude R.

2. Consider a state space that is a tree with branching factor b and maximum depth m. Answer parts a and b by placing X's in the appropriate squares in the table below.
    a. Which of the following search methods might take **<u>time</u>** that grows exponentially with m? (Please places X's in the squares in column a corresponding to the search methods for which you think it is true.)
    b. Which of the following search methods might take **<u>space</u>** that grows exponentially with m? (Please places X's in the squares in column b corresponding to the search methods for which you think it is true.)

	a	B
	Can use **time** exponential in m	Can use **space** exponential in m
Depth-first search		
Breadth-first search		
Best-first search		
Hill climbing search		
A* search		

    c. Can depth-first search ever use more space than A* search?

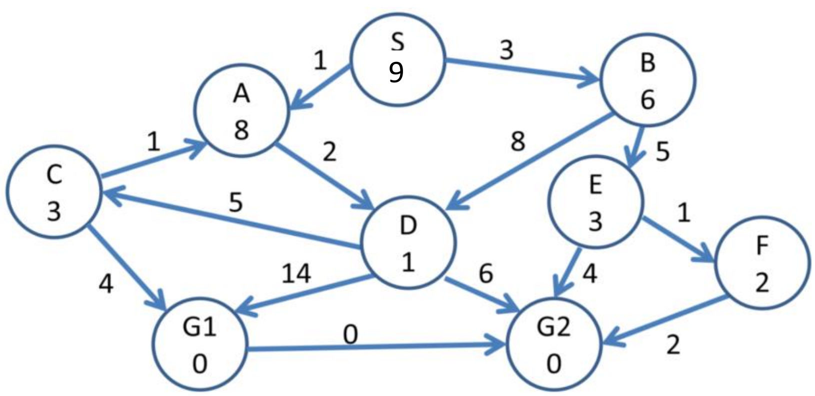
- If your answer is yes, please give an example.
- If your answer is no, please explain why.

3. For each of the following search methods assume the branching factor is b. Give your answer using big-O notation.
   a. What is the worst-case size of the Open list after depth-first search has visited m states?
   b. What is the worst-case size of the Open list after breadth-first search has visited m states?
   c. What is the worst-case size of the Open list after iterative deepening search has cumulatively visited m states?
   d. What is the worst-case size of the Open list after beam search has visited m states? (Assume the beam size is k.)

4. Consider a search problem whose state space may contain cycles or be infinite. Imagine you have an evaluation function f(s) that always assigns to a state a positive integer value that is less than or equal to 100 (where lower values are better).
   Will hill climbing search always halt on such a problem?
   - If your answer is "yes", what is the maximum number of states it will visit?
   - If your answer is "no", please give an example of a search problem where it doesn't halt.

5. Give the smallest example that you can of a search problem with a corresponding h(s) where if you used f(s) = g(s) + h(s) A* would succeed in finding a solution whereas if you used f(s) as the evaluation function for hill climbing search it would fail to find a solution.

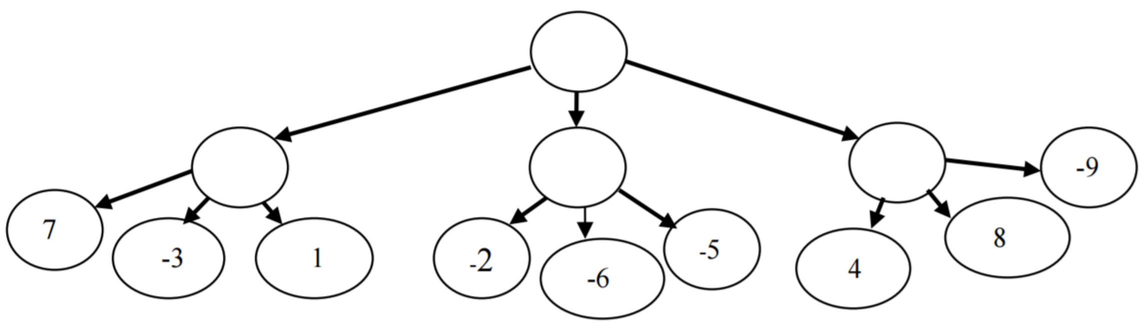
6. Consider the following search space, where S is the initial state and G1 and G2 are goal states, where the cost of the operator that takes you from one state to the next is given along the edge between the states, and where the value of the heuristic evaluation function h applied to the state is the number written inside the state. In cases of ties pick the state that comes earlier in the alphabet.



a. Give the sequence of states that depth-first search visits.
b. Give the sequence of states that iterative deepening visits.
c. Give the sequence of states that hill climbing with f(s) = g(s) + h(s) visits (lower values are better).
d. Give the sequence of states that A* visits if it uses f(s) = g(s) + h(s).
e. Is h(s) admissible?

7. Consider using A* search for a problem where all actions have the cost greater than or equal to 1. Imagine you're using an h(s) that does "one step look-ahead" – namely if after applying one operator to s you are in a goal state h returns 1 otherwise it returns 2. Is h(s) admissible? Please explain your answer.

8. Apply the minimax algorithm to the game tree below, where it is the **opponent's** turn to move next and the leaf nodes are terminal nodes whose values are given in each node in the figure. Process this game tree working left-to-right.



a. Write the values that minimax gives the intermediate nodes inside their circles.
b. Circle the outgoing arc of the root node that represents the move that minimax search would select for this game.
c. Put X's through the nodes that would be pruned by alpha-beta pruning.

9. Is the worst-case time complexity for minimax search and minimax search using alpha-beta pruning the same? Please explain your answer.

10. Give the smallest game tree you can for which alpha-beta pruning would prune at least one node.

11. Give the smallest game tree you can for which alpha-beta pruning would prune at least two nodes that have different parent nodes above it.

12. Resolution is *refutation complete*, in that if KB ⊨ ϕ then if you negate ϕ and convert KB and the negated ϕ to CNF you are guaranteed to be able to generate the empty clause. But how about if you simply start with KB and apply resolution repeatedly until you get

ϕ? Give an example of a KB and ϕ for which KB $\models \phi$ but you cannot prove ϕ directly from KB in this fashion.

13. Convert $\neg ([(P \wedge \neg Q) \Rightarrow (R \vee S)] \wedge T)$ to CNF.

14. Fill in the missing information to make the following application of resolution correct:

$$\neg P \vee \neg T \vee \_\_\_\_\_$$

$$S \vee Q \vee \_\_\_\_\_ \vee U$$

$$\neg P \vee \neg T \vee Q \vee \neg R \vee U$$

15. Consider the following sentence in propositional logic.

$$P \wedge \neg Q \wedge (P \Rightarrow R) \wedge (\neg Q \vee W) \wedge (W \Rightarrow P) \wedge (\neg R \vee W)$$

Show W using resolution.

16. Imagine you are using the hillclimbing approach to find a truth assignment that satisfies a given Boolean formula: You pick an initial random assignment, and at each step you flip one assignment guided by the difference in how many clauses are satisfied by the assignment. Is it possible that this algorithm will fail to find a truth assignment for some satisfiable sentences? If yes, give an example where this happens. If not, explain why not.

17. What is the result if you unify P(F(G(x)),x) with P(F(y),F(G(A)))?

18. What is the result if you unify P(F(G(x)),x) with P(y,F(G(y)))?

19. Imagine you have a scene with three blocks D, E, and F whose colors and relative positions are written in first-order logic:

    Green(D)
    ¬Green(F)
    On(D,E)
    On(E,F)

    Furthermore, you are also given:

    $\forall xyz\ [(On(x,y) \wedge On(y,z)) \Rightarrow On(x,z)]$

    Show using resolution that $\exists x\ \exists y\ [On(x,y) \wedge Green(x) \wedge \neg Green(y)]$.

20. Consider a Markov Decision Process defined as follows:
    - There are just two states, A and B.
    - In either state there are just two actions, 1 and 2.
        - In either state if you do action 1 then with 90% probability you wind up in state A and with 10% probability you wind up in state B.

    - In either state if you do action 2 then with 90% probability you wind up in state B and with 10% probability you wind up in state A.
    - For any action that lands you in state A your reward is 1.0, and for any action that lands you in state B your reward is 0.0.
- $\gamma = 0.5$

  a. Consider a policy Π that always does action 2 in state A and always does action 1 in state B. What is $U^{\Pi}(A)$?
  b. What is $U^{\Pi}(B)$ for this policy?
  c. What is the optimal policy Π* for this problem?
  d. What is U*(A)?
  e. What is U*(B)?

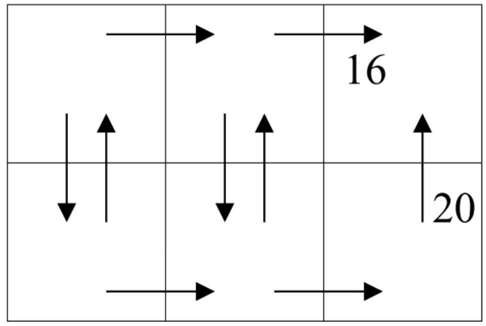
21. Consider the same 4-state Markov decision problem used as a running example in class, only where the states “wrap around” if you go off the edge of the graph – going left from <1,1> takes you to <2,1>, going down from <1,1> takes you to <1,2>, and so on. Everything else about the problem is unchanged. Assume γ=0.7.
    a. Policy Π_{UR} says go Up in states <1,1> and <2,1>, and go right in states <2,1> and <2,2>. What is $U^{\Pi_{UR}}$(<1,1>)?
    b. Do three steps of policy iteration, starting with U_0=0 for all states.

22. There are *N* cities along a major highway that forms a big loop. The cities are numbered 1 through N. If you go clockwise from city *i* you get to city *i*+1 and if you go counter-clockwise you get to city *i*-1, except that counter-clockwise from city 1 puts you in city *N* and clockwise from city N puts you in city 1. You start in city 1. Each day you can either
    - do the STAY action, in which case you’ll be in that city the following day
    - do the CLOCKWISE or COUNTER-CLOCKWISE actions, in which case with probability pi you’ll wind up in the next city in the selected direction the next day, but with probability (1-pi) your car won’t start in which case the city you’re in will be unchanged the next day.

    All even numbered cities give reward $r_i = 1$, and all odd numbered cities give reward r_i=0, if you are in that city at the start of a given day.

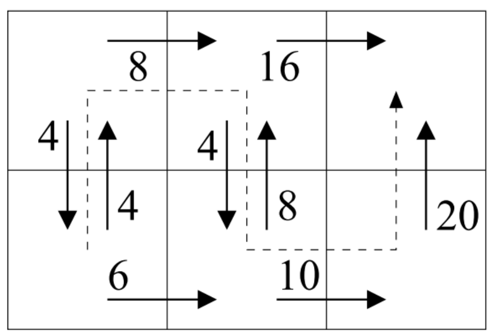
    i. If for all cities pi = 1 and the discount factor $\gamma = 0.5$:
        a. What is the value of $U^{\pi}(1)$ for the policy π that says to execute STAY in all states?
        b. What is the value of $U^{\pi}(N)$ for this policy?
        c. What is the optimal value U(1) for city 1? What policy does it imply?
        d. What is the optimal value U(N) for city N? What policy does it imply?
        e. If N=3: Show the values of π for the first two iterations of policy iteration, assuming that for each state π is initially set to CLOCKWISE.

23. Consider the following Markov Decision Process:



An operator that moves to the right is called R, down is D, and up is U. Each action is deterministic. States range from (1,1) at the bottom left to (3,2) at the top right. (To avoid clutter in the figure they are stated here in text rather than on the diagram.) The reward function between any two states is 0 except that R(<3,1>,U,<3,2>) = 20 and R(<2,2>,R,<3,2>) = 16. If there is no arrow corresponding to R, D, or U in a state then the corresponding action does not apply to that state.

Imagine that partway through Q-learning the values of the Q function are as given in the figure below along the actions arrows:



Thus, for example, Q(<1,1>,R) = 6.

Recall that the Q learning update rule is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(N(s,a)) \times [\ \{r + \max_{a' \in A} \gamma\, Q(s', a')\} - Q(s, a)\ ]$$

Write on the diagram the new values for each Q if you follow the dashed path (5 actions total). Assume $\alpha(n) = 1$ and $\gamma = 0.5$.