1) Suppose you had to write a path finder for a character trying to catch a target in an environment with some static and some moving obstacles. If the target was standing still, would D* be typically more efficient than A*? What if the target was moving? Explain.

2) Unlike A*, Learning Real-Time A* (LRTA*) only expands N states when planning (instead of continuing until the goal is expanded). If the usual assumptions for LRTA* are met, which of the following statements are true? (circle all that apply)

   a) An agent using LRTA* will follow the optimal path to its goal.
   b) An agent using LRTA* is guaranteed to reach its goal.
   c) An agent using LRTA* will never visit a state more than once.

3) Recall that Iterative Deepening A* (IDA*) runs a DFS and prunes states that exceed a maximum f-value. If the goal is not found, the maximum f-value is increased and the process is repeated. Under what conditions is IDA* guaranteed to find the optimal solution? Briefly explain.

4) Compare Iterative Deepening A* (IDA*) and A*. When would and wouldn’t you use IDA*?
5) When using value iteration to solve an MDP with the intention of minimizing the expected cost to the goal what is the update rule for a state's $v$-value? Once value iteration has converged, what is the policy (given a state, what is the best action to take)? Remember in an MDP an action can go to several states according to a probability distribution.

\[ v(s) = \]

\[ \pi(s) = \]

6) When running value iteration to solve an MDP, the Bellman equation is applied to each of the states in some ordering. Then the whole process is repeated until convergence. The order chosen can make a large impact on how many times value iteration runs through all the states. On an MDP with no probabilistic edges (a normal graph), in the worst case, how many times will value iteration run through all the states? What is the best case?

7) What is the main advantage of naïve bayes over n-grams?
8) Can the following procedures be accomplished using a cellular automata rule? If so, provide the rule.
   i) Map Flooding

   ii) Convolution

9) Which of the following games have transpositions (more than one way to reach the same game state). Circle all that apply.
   
   a) Tic-Tac-Toe
   b) Connect 4
   c) Chess
   d) Checkers

10) In the minimax tree below, cross out the nodes that are not evaluated if we are using Alpha-Beta pruning. Also, indicate what the resulting max score is.
11) Run A* on the grid below and fill in the g-values of the cells that would be expanded when A* terminates. The start is marked S, the goal is marked G and black cells are obstacles. Use an 8-connected grid where axis-aligned movements cost 1 and diagonal movements cost 1.4. For the heuristic use max \((dx, dy)\). Use the grid on the top for scratch work and the grid on the bottom for your final answer.
12) Using the given masks, compute the 2D convolution on the map. The “center” of the mask is the element at its center (element in second row, second column). When the convolution tries to access map elements that are outside of map bounds, assume zero values.

Map:

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Mask:

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Convolved Map:
13) Would you use Unity's built-in state synchronization features (option A) or custom RPCs (option B) to share updates on the following events with players in a multiplayer game?
   A door opening: _________
   A character moving: _________
   A character animation changing: _________
   A character dying: _________
   A boulder tumbling down a hill: _________
   A light turning on: _________

14) There is a GameObject in your scene named "Vehicle1" and attached to it is a C# script named "VehicleController" with a public float member variable "desiredSpeed".
   You are writing a separate script and you don't have a reference to the vehicle, the script attached to it, or the desiredSpeed, but you want to access the desiredSpeed for Vehicle1. You can use the following API calls. Use them to write a C# statement that would increment desiredSpeed for Vehicle1 by 1.0f;

15) In level-of-detail-based task management for AI calculations, which of the following might be used as a metric for agent importance (circle all that apply)?
   a) Agent's size
   b) Proximity to player
   c) Agent's visibility
   d) All of the above
16) The output of a kinematic movement system is:
   a) Position and orientation of character
   b) Desired velocity and orientation of character
   c) Actual velocity and orientation of character
   d) Acceleration and angular acceleration of character

17) Wright's Method for scheduling tasks ...
   a) ... is a "greedy" algorithm for scheduling tasks
   b) ... is an optimal algorithm for scheduling tasks

18) Write a formula for dynamic acceleration output for the seek behavior.

Let
P = simulated agent’s position
T = target position
d = T-P = [dx, dy]
M = maximum acceleration
normalize() normalizes a vector and returns the result

A = acceleration output =