**CAP4630 Introduction to Artificial Intelligence**

**Homework 5 (9 pts, Due Nov 14 2022. Firm Deadline)**

[Homework solutions must be submitted through Canvas. No email submission is accepted. Only pdf, word, and txt files are allowed. If you have multiple pictures, please include all pictures in one Word/pdf file. You can always update your submissions before due date, but only the latest version will be graded.]

1. [1 **pt**] Complete following table (yes or no) to compare main differences between Hill-climbing search vs. Best-First search

|  |  |  |
| --- | --- | --- |
|  | Hill-climbing | Best-first search |
| Create a search node? | Yes | Yes |
| Create a search tree? | No | Yes |
| Create a search path? | No | Yes |
| Backtrack search? | No | Yes |
| Use heuristic function? | Yes | Yes |
| Greedy? | Yes | Yes |
| Completeness? | No | No |
| Optimality? | No | No |

1. [**1 pt**] Figure 1 shows a robot navigation field where the robot (red dot) is at location (g,2), and wants to navigate to the destination (green dot) at (g,17). Robot is only allowed to move horizontally or vertically. Use Euclidean distance (straight line distance) as the heuristic, please show the movement of the robot using Hill-Climbing search.
   1. Show the movement (trace) of the robot [0.5 pt]
   2. Can robot reach the destination, why or why not [0.5]

I’m going to assume a=0, b=1, c=2, d=3, e=4, f=5, g=6, h=7, i=8, j=9, k=10, l=11, m=12

Robot is going to go straight to the wall because that will be the shortest path to the goal state. But when it can’t go further it will be stuck because it will have arrived at a local-maxima.

I created an Excel Spreadsheet to calculate the Euclidean Distance for every node (current, above, below, left, right). You can see the data that proves my above statement.

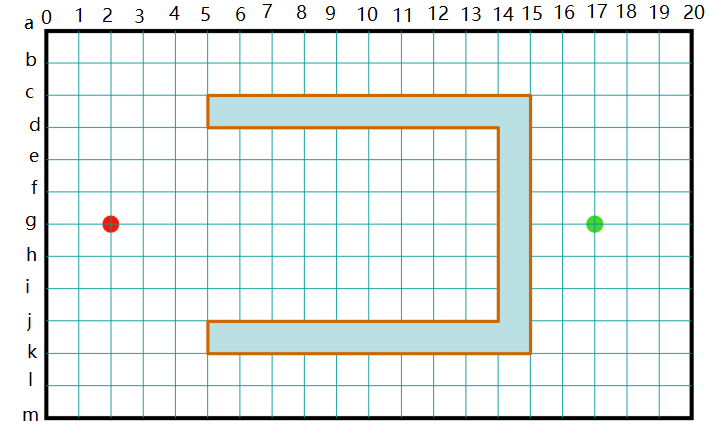


Figure 1: Robot navigation

Table

Description automatically generated

1. **[1 pt]** Describe simulated annealing search [0.25 pt]. How does simulated annealing search behavior when the temperature T is very large vs. T is very small [0.25 pt]. What is main difference between simulated annealing vs. Monte Carlo descent [0.25 pt]. If T=0 at all times, which search method is the simulated annealing search equivalent to [0.25]?
2. Simulated annealing search is based off the process of annealing in metallurgy. This is where the metal is brought to a high temperature which excites the atoms/molecules and allows them to move freely. The temperature is then decreased slowly and this allows the atoms/molecules to arrange themselves in a stable pattern through random motion. Our simulated annealing search works similarly in that it is initially allowed more freedom to explore the graph but as it explores, begins to settle down into a more stable path.
3. When T is high, the algorithm is allowed to move in all directions. It can choose paths that are not optimal and this gives it freedom to move about the graph. As the search progresses, the temperature decreases and the algorithm begins to choose more optimal paths (the variance allowed between optimal and nodes shrinks). When the Temperature reaches 0, the algorithm will only choose an optimal path.
4. The main difference between Simulated Annealing and Monte Carlo Descent is the random restarts that are used in Monte Carlo Descent. Both will explore larger portions of the graph and escape local maxima, but they achieve this in different manners.
5. If T=0 at all times, than the Simulated Annealing Search will only choose optimal paths and this is the same as Hill-Climbing Search.
6. **[1 pt]** Figure 2 shows the current layout of a tic-tac-toe game board, and MIN (“O”) is going to make the next move. Please draw the remaining states of the game tree [0.5 pt], and explain what is the best move for MIN [0.5 pt].

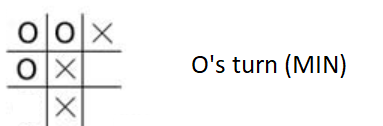


Figure 2

Letter

Description automatically generated

The best move for ‘O’ is any move that lowers the score for ‘X’. The obvious choice is the branch furthest to the left in the image above. The other two scores don’t change the score and leave ‘X’ open for the maximizing score and to win. ‘O’ can still win in the other two branches, but that requires ‘X’ to make a mistake and we want to assume that ‘X’ will not make a mistake and will be trying to maximize its score and to win. So the best choice is the branch furthest left.

1. **[1 pt]** Given a two-player game tree in Figure 3, starting from the root node, please list the nodes in the order they are visited by the Mini-max search [0.5 pt], please show utility values of every node [0.5 pt].

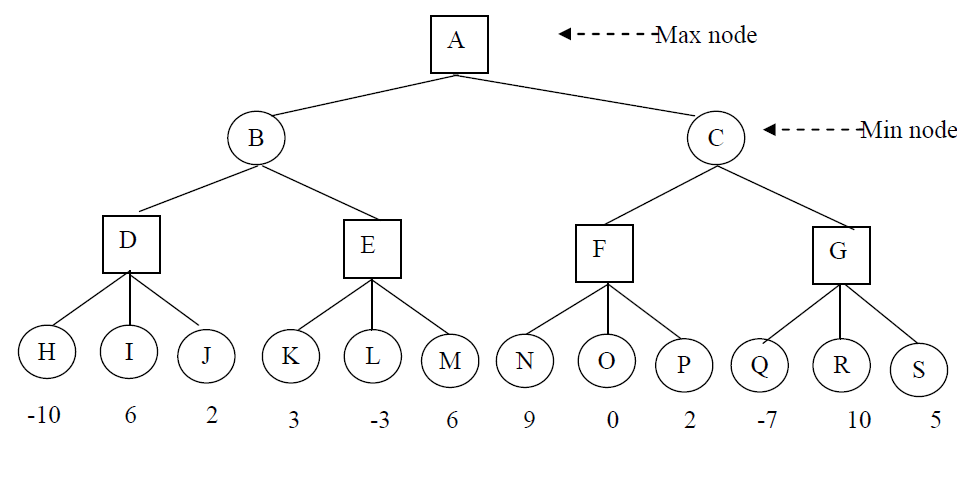


Figure 3

Diagram, engineering drawing

Description automatically generated

The Final Path will be: A -> C -> F -> N

The Utility Values are determined by the whether the previous node is a min or max node. If it is max, it will choose the highest value from the next nodes available. If it is min, it will choose the lowest. i.e., node F will choose because it is a max node, and it will choose the largest value from the nodes following it. Node C will be 9 because it can choose from F:9, or G:10. F is lower, so it chooses that value.

Utility Values: A:9, B:6, C:9, D:10, E:6, F:9, G:10 and the others are already given in the problem.

1. **[1 pt]** Given a two-player game tree in Figure 3, starting from the root node, please list the nodes in the order they are visited by α-β pruning [0.25 pt]. Please show the utility values of every node [0.25 pt], and report nodes which are pruned by α-β pruning [0.5 pt]

Diagram

Description automatically generated

The order in which nodes are visited: H, I, J, D, K, L, M, E, B, N, O, P, F, Q, R, G, C, A

Please note that node S was skipped or “pruned” from α-β pruning because when the algorithm finds the utility value of 10 in node R, it is already larger than the other option at the same level (which is node F with a utility value of 9). Therefore it is not necessary to expand node S to see what its utility value is. The utility values are shown in the image above.

1. **[1 pt]** Given a two-player game tree in Figure 4, starting from the root node, please list the nodes in the order they are visited by α-β pruning (0.25 pt). Please show the

utility values of every node [0.25 pt], and report nodes pruned by α-β pruning [0.5 pt].

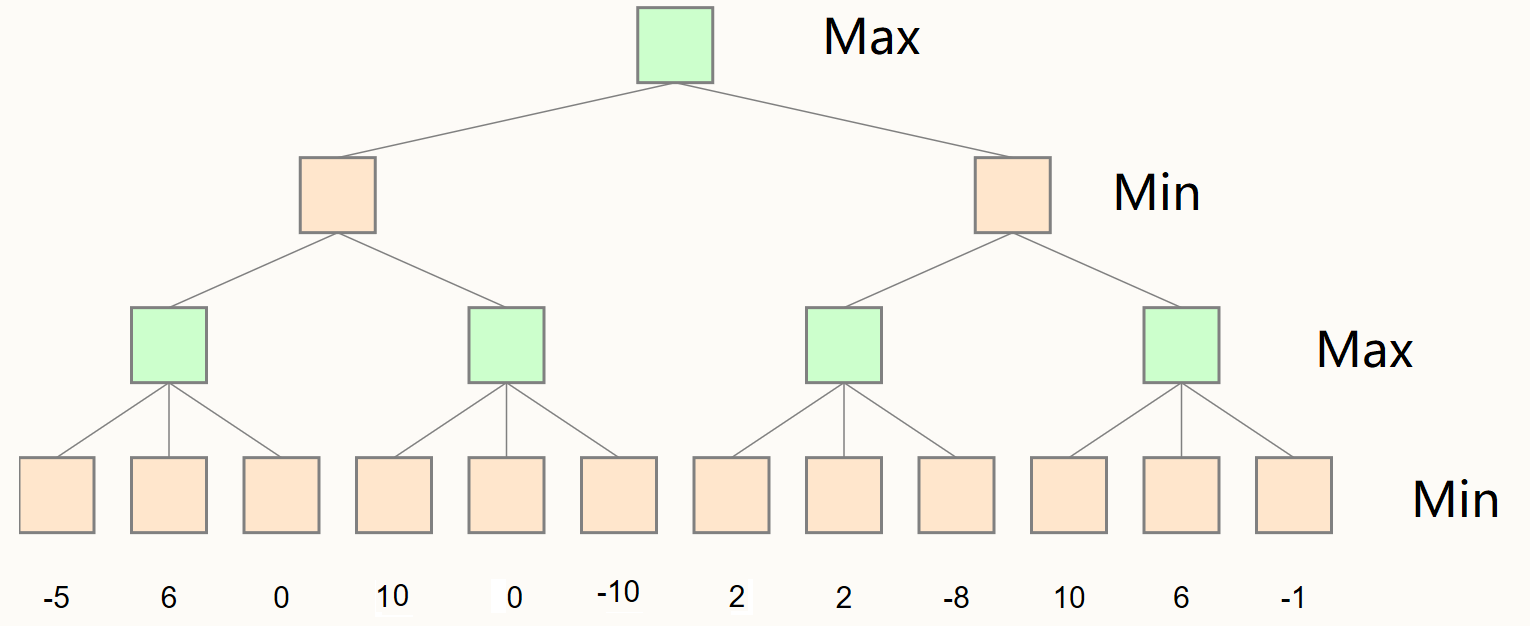


Figure 4

Diagram

Description automatically generated

The order in which the nodes are visited: H, I, J, D, K, E, B, N, O, P, F, Q, G, C, A

I labeled each node alphabetically. Please note that nodes L, M, R, and S are skipped or “pruned” from α-β pruning. It is not necessary for the algorithm to expand those 4 nodes because it already found a maximum higher than the node adjacent to it. The utility values are shown in the image above.

1. **[1 pt]** Given a two-player game tree in Figure 5, starting from the root node, please list the nodes in the order they are visited by α-β pruning (0.25 pt). Please show the utility values of every node [0.25 pt], and report nodes pruned by α-β pruning [0.5 pt].

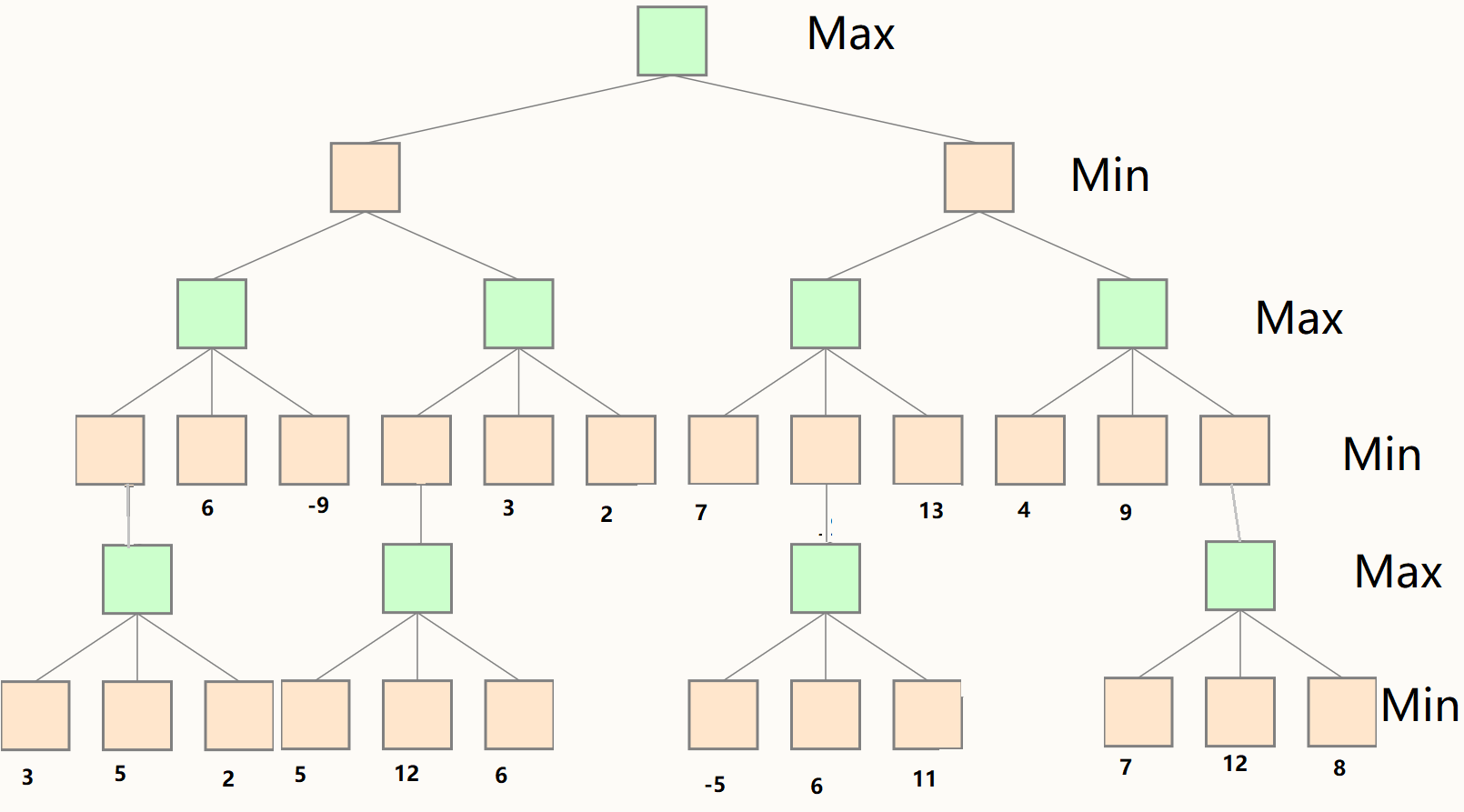


Figure 5

Diagram

Description automatically generated

The order in which the nodes are visited: X, Y, Z, T, H, I, J, D, AA, AB, AC, U, K, E, B, N, AD, AE, AF, V, O, P, F, Q, R, AG, AH, AI, W, S, G, C, A

I labeled each node alphabetically. Please note that nodes L and M are skipped or “pruned” from α-β pruning. It is not necessary for the algorithm to expand those 2 nodes because it already found a maximum higher than the node adjacent to it. The utility values are shown in the image above.

**Question 9 [1 pt]** The Minmax Decision Playing Tictactoe [Notebook, html] and Alphabeta Pruning Playing Tictactoe [Notebook, html] posted on Canvas show two programs playing Tic Tac Toe game against a computer agent. Use Notebook as the skeleton code, validate and compare following settings and results.

* 1. Play Tic Tac Toe Minmax for 10 times. Report average computer thinking time (which is the time required for Minmax algorithm to find solutions) and number of times you have won/lost/tied [0.25 pt]. Explain why it is hard to win the game against computer [0.25 pt]
  2. Play Tic Tac Toe Alphabeta for five time. Report average computer thinking time (which is the time required for Alpha Beta Pruning to find solutions) [0.25 pt]. Compare time required for Minmax and Alphabeta pruning, explain why Alphabeta pruning is quicker [0.25 pt].

I included a table below that displays the different attempts for each version of the game and the average computer thinking times.

It is hard to win the game against the computer because the computer is evaluating every possible state after each move and choosing the move that is most optimal for it to win.

After comparing the computing time for each algorithm, it is evident that Alphabeta is much faster. This is because the computer is “pruning away” states that it does not need to evaluate. It is only focusing on those necessary to win.

