CS347 SP2004 Exam 3

This is a closed-book test. The only item not supplied that you are allowed (and required) to use, is a pen or pencil. Mark every sheet of paper you use with your name and the string “cs347sp2004 exam3” (omittance, even if it is partial, will be penalized at 1 point per sheet). If you are caught cheating, you will receive a zero grade for this test. The max number of points per question is indicated in square brackets after each question. The sum of the max points is 105, but the max exam score is capped at 100. You have exactly 50 minutes to complete this test. Good luck!

1. Consider the following list of games:
   - Chess
   - Tic-tac-toe
   - Stratego
   - Backgammon
   - One hand of Poker with the deck shuffled before the start of the hand but not during
   - Multiple hands of Poker with the deck shuffled before each hand
   (a) Indicate which of these games are full-information and which are partial-information. [3]
   (b) Indicate which of these games are deterministic and which are stochastic. [3]

2. Here are three conceptual questions regarding Samuel Mulder’s presentation:
   (a) Explain in one sentence the difference between a trajectory search algorithm and a population search algorithm. [2]
   (b) Give an example of each. [2]
   (c) Is IDA\(^*\) Search a trajectory search algorithm, a population search algorithm, or neither? Explain your answer briefly! [3]

The next three questions are about the following adversarial “chance” tree.

3. Calculate the EXPECTMINIMAX values for nodes B, C and D in the above adversarial “chance” tree. Show your calculations! [3]

4. Which action will MAX choose, \(a_1\), \(a_2\), or \(a_3\)? Explain your answer! [2]

5. If the utility values given for MIN were multiplied with a positive constant \(c\), which action would MAX then choose? Explain your answer! [2]
6. Explain briefly how the idea of alpha-beta pruning can be applied to game trees with chance nodes. [5]

The remainder of the questions are about the following state space graph. Let A be the start state and C the goal state. The edge labels indicate step-cost, the vertex labels contain the node identifier in the form of a letter. Heuristic \( h(node) \) is defined as the Manhattan distance between \( node \) and the goal state. The order in which successors are generated is counterclockwise, ending at exactly 9 o’clock. Example: E generates first H, then F, then B, and finally D. When sorting by f-value, nodes with equal f-value are ordered such that the earlier a node is generated, the higher its priority. Nodes already on the open list have higher priority than newly added nodes with equal f-value.

7. Give the execution trace for Uniform Cost Graph Search (UCGS). [15]

8. Is UCGS optimal for this problem? Explain your answer! [5]

9. Give the execution trace for A* Graph Search (A*GS) employing heuristic \( h \). [15]

10. Is for this problem \( h \) admissible? Explain your answer! [5]

11. Is for this problem \( h \) consistent? Explain your answer! [5]

12. Is A*GS employing heuristic \( h \) optimal for this problem? Explain your answer! [5]


14. Give the execution trace of Iterative Deepening A* (IDA*) Search employing heuristic \( h \). [20]

15. Is IDA* employing heuristic \( h \) optimal for this problem? Explain your answer! [5]