CS 5100

Homework 7

Marking Guidelines

(50 Marks)

Answer the following questions

Question 1: (10 Marks)

Write a program in a programming language of your choice for solving the n-queens problem. Run it with 8 queens, and then try it with 100 queens. How well does it perform? Could your program find a solution for 1,000,000 queens? If not, why not? If so, what optimizations have you used to make that possible?

Criteria	**Marks**
Program correctly solves the 8-queens problem	3
Program attempts and reports result for 100 queens	2
Performance discussion for both cases (8 and 100 queens)	2
Attempt or discussion for 1,000,000 queens	1
Reasonable explanation of why it does/doesn’t scale (complexity, memory, algorithm, etc.)	1
Discussion/implementation of optimizations (if any)	1

Question 2: (10 Marks)

Write a program that can solve arbitrary cryptographic problems. Add heuristics to your implementation to make it more efficient. What limitations does your program have?

Criteria	**Marks**
Program can solve at least basic cryptographic problems	3
Attempt to add heuristics or efficiency improvements	2
Explanation/discussion of heuristics added	2
Identification of limitations (scalability, types of ciphers, etc.)	2
Code is clear and commented (if submitted)	1

Question 3: (5 Marks)

Discuss the current state of the art of game-playing computer systems in relation to the following games: chess, checkers, Go, bridge, Othello, tic-tac-toe. What advances are likely in the near future? Do you believe there are any fundamental limitations?

Criteria	Marks
Clear discussion for at least four games (chess, Go, etc.)	2
Mention of recent advances or technologies (e.g., AlphaZero, RL, deep learning)	1
Thoughtful speculation about near-future advances	1
Discussion of fundamental limitations (computational, theoretical, etc.)	1

Question 4: (5 Marks)

Explain why the alpha–beta procedure will always generate the same answer as Minimax without pruning. Why is it useful?

Criteria	Marks
Correct explanation why alpha–beta and minimax produce the same result	2
Explanation of the usefulness/efficiency of alpha–beta pruning	2
Clarity and correctness of explanation	1

Question 5: (20 Marks)

a) Write an algorithm in pseudo-code, or a programming language of your choice, that evaluates a position in tic-tac-toe. Your static evaluator should give 0 for a drawn position, 1 for a won position for crosses, -1 for won position for noughts. (6 Marks)

Criteria	Marks
Pseudocode or code for evaluator	2
Evaluator correctly distinguishes between win/loss/draw	2
Returns correct values (1, -1, 0) as specified	2

b) Implement a Minimax algorithm using your static evaluator for tic-tac-toe and write a simple program that plays the game. Have the program output how many nodes in the game tree it had to examine as well as its choice of move. (14 Marks)

Criteria	**Marks**
Correct Minimax implementation (working code)	6
Uses the evaluator from (a) properly	2
Program can play the game and make moves	2
Outputs number of nodes examined	2
Code is clear, well-commented, and readable	2