**Steps for Completion:**

1. **Understand the Knight's Tour Problem (KTP):**
   * Read and summarize the key points from the paper *Heuristic Strategies for the Knight Tour Problem*.
   * Formally describe the KTP:
     + A knight moves in an L-shape: two squares in one direction and one square perpendicular, or vice versa.
     + The goal is to visit every square on the chessboard exactly once.
2. **Classify the Problem’s Environment:**
   * Clearly specify the characteristics of the environment (static/dynamic, discrete/continuous, deterministic/non-deterministic, etc.).
3. **Tree Search Framework:**
   * Implement the **Tree-Search Algorithm** from the provided image as the core framework for all search methods.
4. **Search Methods Implementation:** Implement four search algorithms:
   * **Breadth-First Search (BFS):** Use a queue for managing the frontier.
   * **Depth-First Search (DFS):** Use a stack for managing the frontier.
   * **DFS with Node Selection Heuristic h1b:** Implement the heuristic h1b from the paper.
   * **DFS with Node Selection Heuristic h2:** Implement the heuristic h2 from the paper.
5. **Input and Output:**
   * Design the program to accept:
     + n (size of the board),
     + a-d (chosen search method),
     + t (time limit).
   * Ensure outputs include:
     + Search method and time limit.
     + Return status (solution found, no solution, timeout, or out of memory).
     + Solution path, if found.
     + Execution time.
     + Number of nodes expanded.
6. **Testing and Analysis:**
   * For each search method (a-d) and board sizes (8, 16, 32, 41, 52), report results with a 15-minute time limit.
   * For method (d), determine the largest board size solvable within 15 minutes.
7. **Design Document and Comments:**
   * Write a detailed design document covering:
     + Problem description and environment classification.
     + Design choices for the implementation.
     + Explanation of heuristics and their impact.
   * Add clear and descriptive comments in your code.