**Machine Learning and Smart Systems**

**Assignment 01 – Implementing Search Algorithms**

**Solving the Rubik’s Cube using Search Algorithms (DFS, BFS, and A\*)**

**Objective:** Design and implement three different approaches to solve the Rubik’s Cube problem using uninformed search techniques:

1. Depth-First Search (DFS)
2. Breadth-First Search (BFS)
3. A\* Search

**Instructions:**

**Design Phase:**

1. **Choose a programming language:** Use any programming language you are comfortable with. However, Python is recommended due to its rich ecosystem for algorithms and problem-solving.
2. **Understand the Rubik’s Cube Problem**: Familiarize yourself with the Rubik's Cube puzzle, including its state-space representation and possible moves.
3. **Develop algorithms for solving the Rubik's Cube**:
   1. Implement DFS, BFS, and A\* search algorithms to find a solution to the Rubik’s Cube.
   2. Use DFS and BFS for exhaustive search methods, while A\* should employ a heuristic-based approach for better performance.
4. **Document the design:** 
   1. Discuss the selection of each algorithm and any optimization techniques you might have implemented.
   2. Consider time and space complexities for each approach and discuss any trade-offs.

**Implementation Phase:**

1. Implement the algorithms:
   1. Develop the complete solutions using DFS, BFS, and A\* search.
   2. For A\*, implement an admissible heuristic function to improve the search process.
2. Test your solutions:
   1. Test the three approaches with the Rubik’s Cube at different difficulty levels (e.g., 2x2x2, 3x3x3, 4x4x4, 5x5x5, and 6x6x6).
   2. Record the execution time for each approach and document any edge cases or issues.
3. Record execution time: For each case (e.g., different cube sizes or scrambling), log the time taken and memory required to solve the cube.

**Analysis and Comparison:**

1. Compare the three algorithms:
   1. Analyze the time and space efficiency of DFS, BFS, and A\*.
   2. Discuss which algorithm performs best under different conditions (e.g., difficulty of the initial state, depth of solution).
   3. Compare their effectiveness, particularly focusing on optimality, completeness, time complexity, space complexity, and real-world applicability.

**Report: (A sample pdf report for N Queens Problem is attached for your reference)**

1. Document the design, implementation, and results:
   1. Use the Overleaf report template provided to you. You may also write your report in Microsfot Word.
   2. Include the following sections:
      1. **Introduction:** Overview of the Rubik’s Cube problem and the search algorithms.
      2. **Literature Review:** Discuss existing solutions or related work with how other scientists or studies have addressed this problem.
      3. **Methodology**: Detailed explanation of the DFS, BFS, and A\* algorithms, specific to your implementation.
      4. **Results**: Present the test cases, performance data (time taken space taken optimality completeness for different scenarios), and comparisons.
      5. **Analysis**: Insightful discussion of the results.
      6. **Conclusion**: Summarize your findings and provide a final recommendation.
      7. **References**: Cite relevant sources used in your research especially for the Introduction and Literature Review Sections.

**Deliverables:**

1. **Code Files**: Submit the complete code for all three algorithms (DFS, BFS, A\*) as a zipped folder (if you submit report in Word file) or in a separate subfolder in Overleaf repository alongwith writeup.
2. **Report**: Submit the report in either 1. word and PDF format, OR 2. Overleaf editable link (check that it is accesible by first checking it from another account before submitting) and PDF format . The report should be structured according to the Overleaf template and should include figures, tables, and charts where necessary. See the attached sample report.
3. **Overleaf Report Repository**: Use this Overleaf repository for the report structure: [Overleaf Rubik's Cube Report Template](https://www.overleaf.com/read/example).

**Evaluation Criteria**:

1. **Correctness of Implementations**: The solutions should correctly solve the Rubik's Cube for each test case (e.g., 2x2x2, 3x3x3).
2. **Design Documentation**: The clarity, depth, and professionalism of your design documentation.
3. **Analysis of Results**: The insightfulness of your performance comparison and complexity analysis.
4. **Quality of Report**: The overall quality of the report, including adherence to the Overleaf template, the professionalism of figures and tables, and completeness.
5. **Innovation**: Any novel ideas or optimizations you implement in the algorithmic design.

**Rubrics for Evaluation**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criteria | Excellent (5) | Good (4) | Satisfactory (3) | Needs Improvement (2) | Unsatisfactory (1) |
| Correctness of Implementation | All algorithms work as expected, correctly solving Rubik’s Cube for various cases | Most algorithms work correctly with minimal bugs | Some algorithms work, but there are major errors | Only one algorithm works or has severe issues | No algorithms work |
| Design Documentation | Clear, comprehensive, and well-organized design considerations | Clear documentation with minor missing details | Basic design explanations, lacking depth or clarity | Minimal documentation with major gaps | No documentation provided |
| Analysis of Results | Thorough, insightful analysis with comparisons based on multiple test cases | Good analysis with comparison but lacks depth | Basic analysis with limited insights or comparisons | Analysis is incomplete or missing | No analysis provided |
| Quality of Report | Report is professionally written, follows the Overleaf template, includes quality visuals and proper citations | Report follows the template well but lacks detail or quality visuals | Report is functional but poorly organized or lacks detail | Report is incomplete or poorly structured | No report submitted |
| Quality of Figures, Tables and Literature | Professional figures, Well drawn tables and recent literature works (post 2022) cited for the report | Some figures and tables not as good or some earlier than 2023 citations used. | Few figures and tables are good or few later than 2022 citations used. | Figures and tables are of poor quality, unreadable. All citations are from older times. | No Citations done or no Figures or Tables drawn |

**NOTE:** Please note that you may use CHATGPT or similar AI tools for this solution development as well as for writing the report. Make sure that you understand each term, all text, figures and tables are in sync, and you should not share your solution/write up/idea with other students.