**Solution Summary**

The objective of this project was to develop an efficient solver for nonogram puzzles, which include both rectangular and hexagonal grids. Nonograms are logic puzzles where players color cells in a grid according to given clues, forming a picture. The main challenge lies in converting the puzzle into a Boolean Satisfiability (SAT) problem, solving it using a SAT solver, and decoding the solution back into a grid format.

Two different approaches were implemented to encode the nonogram as a SAT problem:

**Basic Encoding (Approach 1):** This straightforward method directly translates the clues into SAT clauses. Each clue for rows and columns is converted into a set of constraints that the SAT solver must satisfy.

**Compact Encoding (Approach 2):** This approach introduces additional variables and more compact representations of the constraints. The goal was to reduce the number of variables and clauses, aiming to improve the performance of the SAT solver, especially on larger puzzles.

These approaches were chosen to explore the trade-offs between simplicity and efficiency in SAT encoding.

**Key Insights and Design Decisions**

**SAT Problem Encoding**

The primary design decision involved how to encode the nonogram puzzle into SAT clauses. Initially, the basic encoding was implemented, where each block of the same color in the clues was directly mapped to a set of variables representing the grid cells. While simple to implement, this method led to a large number of clauses, particularly for larger grids or more complex clues.

To address this, we developed a compact encoding that reduced the total number of clauses by introducing helper variables. These variables allowed for a more efficient representation of the constraints, particularly for cases where multiple possible block placements needed to be considered. This encoding leveraged the SAT solver's ability to handle complex logical relationships more efficiently.

**Handling Hexagonal Grids**

Another significant decision was how to handle hexagonal grids, which differ from rectangular grids in that they have three possible directions (horizontal, diagonal down-right, diagonal down-left). The encoding was adapted to ensure that all three directions were properly considered, and constraints were generated for each direction based on the clues.

**Efficiency Analysis**

To evaluate the efficiency of the two approaches, we measured several key metrics:

**Execution Time:** The time taken by each approach to solve nonograms of varying sizes and complexities.

**Number of Variables:** The total number of variables generated in the SAT problem, which impacts the solver's performance.

**Number of Clauses:** The number of clauses in the SAT problem, which directly correlates with the problem's complexity.

**Results**

The following plot illustrates the performance comparison between the two approaches across different nonogram sizes:

Include a plot here comparing execution times, number of variables, and number of clauses between the two approaches.

**Discussion**

From the results, it is evident that Approach 2 (Compact Encoding) outperforms Approach 1 (Basic Encoding) as the size and complexity of the nonogram increase. The reduction in the number of variables and clauses in Approach 2 leads to significantly faster solve times. However, the compact encoding requires a more sophisticated understanding of SAT solvers and logical problem representation.

**Conclusion**

The nonogram solver developed in this project successfully solves a wide range of puzzles using SAT solvers. The comparison between the basic and compact encoding approaches highlights the importance of efficient problem representation in SAT-based solutions. While the compact encoding approach offers better performance, it also introduces additional complexity in the encoding process.