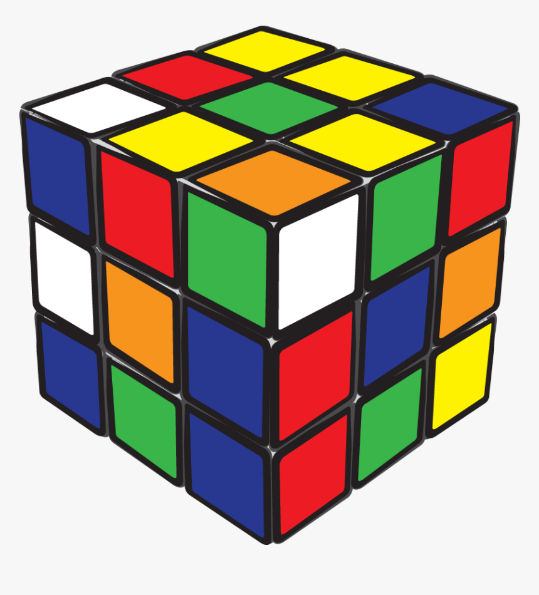
**Artificial Intelligence**

**Solving Rubik’s Cube using Iterative Deepening Search & A\* Algorithm**



# Introduction:

Rubik's Cube is a 3-D combination puzzle invented in 1974 by Hungarian sculptor and professor of architecture Ernő Rubik. Originally called the Magic Cube, the puzzle was licensed by Rubik to be sold by Ideal Toy Corp. in 1980 via businessman Tibor Laczi and Seven Towns founder Tom Kremer and won the German Game of the Year special award for Best Puzzle that year. As of January 2009, 350 million cubes had been sold worldwide, making it the world's top-selling puzzle game. It is widely considered to be the world's best-selling puzzle.

In this assignment, we were required to solve a Rubik’s Cube by using an uninformed search called Iterative Deepening Search (IDS) and an informed search called A\* algorithm that uses a heuristic to guide the search pattern. Initial and Final states of the cube are provided via a file that is loaded into the program as it starts and solves the cube using both algorithms. Both algorithms are implemented in the same cpp file.

# Methodology:

In Iterative Deepening, we use both DFS and BFS techniques to traverse a tree depth by depth. Initially, depth is initialized to zero and then we compare all possible nodes at given depth and traverse them using a combination of DFS and BFS using recursion until the goal is found or the depth level exceeds the provided limit. It is an exhaustive approach in terms of time and space complexity.

Face Descriptions:

* Front face=0
* Top face=1
* Back face=2
* Bottom face=3
* Right face=4
* Left face=5

Each face has two rotations, clockwise and anticlockwise.

In A\* algorithm, we use a heuristic function to guide the search of the nodes until the goal is found or the queue holding nodes becomes empty. The heuristic function takes the current state of the cube and compares all faces index by index excluding the same color indexes with the goal state of cube. Manhattan distance is calculated for each face, summed up and divided by 12 as each rotation moves 12 indexes.

# Results (Test Cases):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Depth | Nodes Traversed | | Time (in seconds) | |
|  | IDS | A\* | IDS | A\* |
| 1 | 11 | 2 | 1 | 1 |
| 2 | 103 | 5 | 1 | 1 |
| 3 | 889 | 7 | 1 | <1 |
| 4 | 4660 | 9 | 1 | <1 |
| 5 | 95145 | 25 | 20 | 3 |
| 6 | 909505 | 40 | 197 | 10 |

# Conclusion:

As it is evident from the Results mentioned above, A\* performs exceptionally well at depths of larger magnitude than IDS. A\* traverses minimum nodes and takes much less time than IDS.