Report

**Introduction**

In this assignment we were assigned to to find path between rubik cube states’ tree to the goal state. For this we were allowed to use2 algorithms. A\* and Iterative Deepning Algorithm. A\* Algorithm did this job efficiently, by keeping updated queue.

**Methodology**

First of all we made relevant classes of Cube, Phase, Node, Helper. Helper classes contain functions which we needed often. Rubik Cube contain functions like of file reading of cubes, produce child states of cube (quite brain consuming), operators and copy functions. A Cube has 6 phases pointers in an array. Tree class contain both functions of A\* search and Iterative Deepening Search. A\* performed better as compared Iterative Deepening

**Results**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | 1 rotation | 3 rotation | 5 rotation | 7rotation | 9 rotation |
| IDDFS | 0.0 | 0.008 | 0.29 | 3.5 | 245 |
| A\* Search | 0.00 | 0.002 | 0.10 | 1.30 | 19.27 |

**Conclusion**

Comparing above performances, A\* algorithm took less time and less number of steps to reach the goal state of the cube presented in the text file. Also it used better heuristic function to predict which path is more near to the goal.

**Heuristic Function Used:**

**We calculated Manhattan Distance i.e of 3 cordinates x, y, z**