Introduction

The Rubik's Cube is an important test domain for heuristic search that has 1019 possible states, making it impossible to store in memory.In order to find a solution to it we need to some sort of search which does not expand all the possible states because we cannot store all the possible states due to memory issue.

Two of the search algorithms which i used are iterative-deepening which is informed search strategy and one informed search strategy named A* .

Methodology:

In this study, two methods were used to solve scrambled rubik's cube. It was done using Iterative Deepening Search (IDS) and A* . Heuristic was used for A*. All methods were able to solve cube. However, time to solve cube increased exponentially with steps to solve the cube. A* with Heuristics solved cube the quickest.

There are 12 possible moves(6 Clockwise and 6 Anticlockwise) which can be applied on every state of a cube. These are coded as functions in the code with clockwise rotation function and anticlockwise rotation function.

iterative-deepening :

In iterative deepening I used a depth first search strategy which goes from depth 0-10.

The dfs function i used is a recursive one. It uses the rotation functions to generate new states of a given state and sends recursive calls to these new states until a solution is found . It uses many helper functions like compare two cube e.t.c.

A*:

A* search algorithm is implemented with an iterative function. It uses many helper functions like compare two cube e.t.c. Furthermore, it uses a Heuristic. The algorithm uses a priority que based on the heuristic value to tell it which state to expand first .

Heuristic:

The heuristic used is very simple. It compares each piece on a cube with the corresponding goal state cube and if it's a mismatch it increments the total mismatches in the whole cube and returns the total mismatches in the cube.

.

Results

Both algorithms were able to solve the cube. IDS time increased rapidly with depth of solution. A* performed quite faster as compared to IDS.

Depth of goal (d)	Search cost (nodes generated)	Search cost (nodes generated)	Time taken						
	IDS	A* (with heuristic)					IDS	A*	
1	12	12					10ms	10ms	
2	12^2 + 12	36					20ms	<1	
3	12^3 + 12^2 + 12	72					20ms	<1	
4	12^4 + 12^3 + 12^2 + 12	180					80ms	<1	
5	12^5 + 12^4	1428					690ms	<1	

	+ 12^3 + 12^2 + 12								
6	12^6 + 12^5 + 12^4 + 12^3 + 12^2 + 12	76680					7s	Got stuck	

Above 7 dfs got stuck.

Conclusion

According to the findings after trying with heuristics we found that the heuristic is the best solution as it finds the solution in less time as compared to others.

To conclude, both IDS and A* can be used to unscramble. But A* with a good Heuristic Is less time and memory consuming .