Rubik's Cube

This code defines a `cube` class that can represent a Rubik's cube. The class has several methods to manipulate and solve the cube, such as `norm()` to normalize the cube to a fixed position, `isSolved()` to check if the cube is solved, `clone()` to create a copy of the cube, `applyMove()` to apply a move to the cube, `shuffle()` to shuffle the cube, `print()` to display the current state of the cube, and `formatPrint()` to display the current state of the cube with a list of moves applied. The `__lt__` method is also defined to enable comparison between `cube` objects.

Additionally, various algorithms are implemented to solve the Rubik's cube.

The breadth first search uses a queue. A cube state is taken from the queue and all of its possible children are added to the queue. This is repeated till a solution is found.

The IDS approach keeps implementing depth limiting search (0 onwards) till the depth limit is reached or a solution is found.

A* uses a priority queue to implement the Manhattan distance heuristic discussed in the problem.

Analysis of Time & Space Complexity with Comparison:

- For depth-first search, with iterative deepening
  - The runtime here is simply just O(D(V+E)) since we are taking the depth limit, D, into V + E which essentially is the same as the original O(V+E) but with this added parameter. Adding a depth limit approach here saves us the trouble of exploring each node unnecessarily and the potentially long route of not terminating fast enough. Limiting the depth allows us to find a slightly more optimal path to the solution. The space complexity of this would be O(V) since we are only considering the vertex at any given iteration.

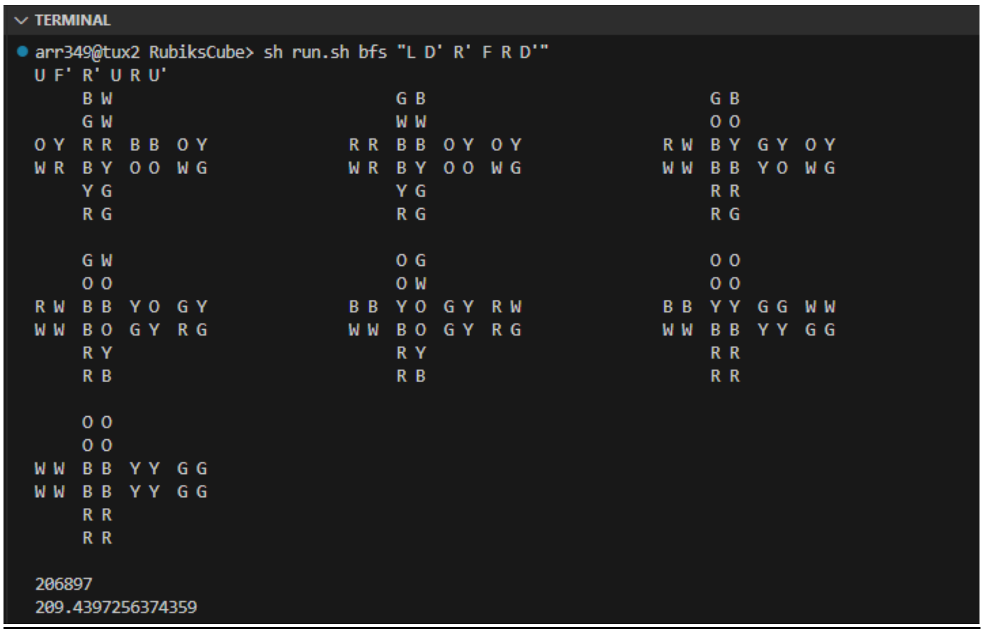
- For breadth-first search
  - The time complexity of this algorithm is O(b^d), where b is the branching factor (number of possible moves) and d is the depth of the solution (number of moves required to solve the cube from the initial state).

  - The space complexity of this algorithm is O(b^d), where b is the branching factor and d is the depth of the solution. This is because the algorithm maintains a list of open nodes, which can contain up to b^d nodes in the worst case, and a set of closed nodes, which can also contain up to b^d nodes in the worst case. Additionally, each node in the open list can have a path of up to d moves stored in memory.

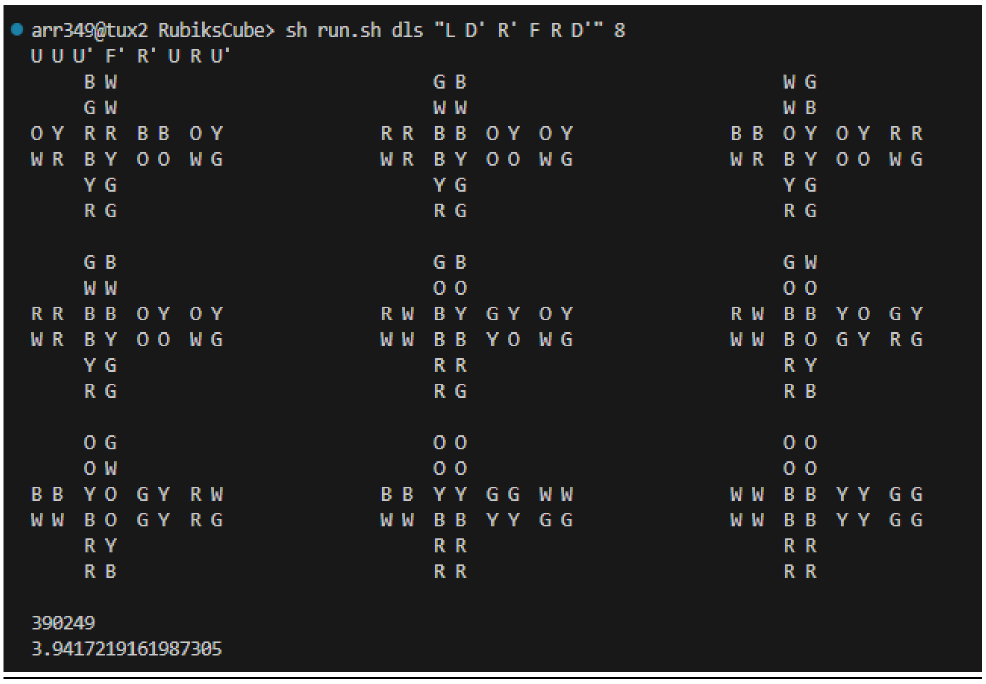
  Heuristic Used for A*

  The Manhattan distance heuristic estimates the cost of reaching the goal state from the current state by calculating the sum of the distances between the current and goal state for each piece of the cube. The distance between two pieces is measured as the sum of the distances between their current and goal positions along each of the three dimensions.

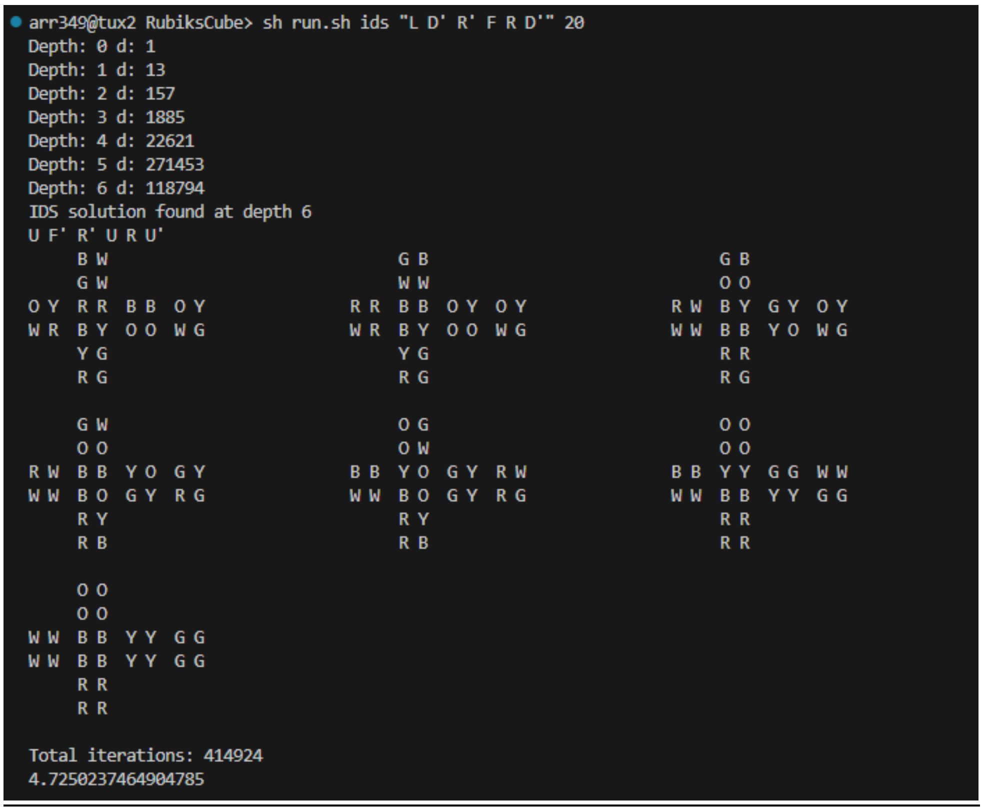
Screenshots



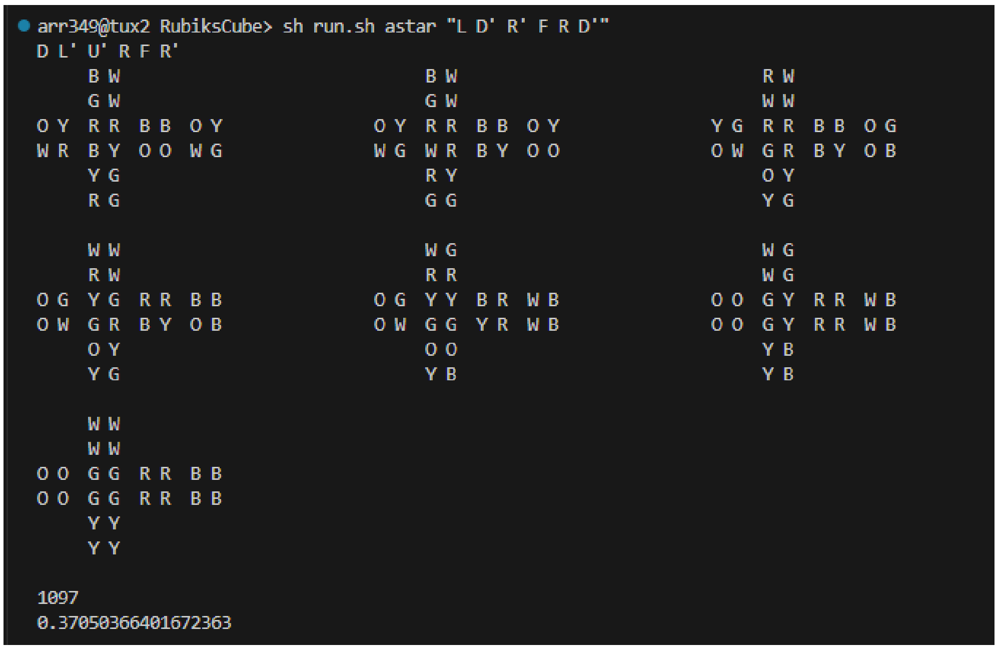
```
● arr349@tux2 RubiksCube> sh run.sh bfs "L D' R' F R D'"
U F' R' U R U'
     B W                              G B                              G B
     G W                              W W                              O O
O Y  R R  B B  O Y               R R  B B  O Y  O Y               R W  B Y  G Y  O Y
W R  B Y  O O  W G               W R  B Y  O O  W G               W W  B B  Y O  W G
     Y G                              Y G                              R R
     R G                              R G                              R G

     G W                              O G                              O O
     O O                              O W                              O O
R W  B B  Y O  G Y               B B  Y O  G Y  R W               B B  Y Y  G G  W W
W W  B O  G Y  R G               W W  B O  G Y  R G               W W  B B  Y Y  G G
     R Y                              R Y                              R R
     R B                              R B                              R R

     O O
     O O
W W  B B  Y Y  G G
W W  B B  Y Y  G G
     R R
     R R

206897
209.4397256374359
```



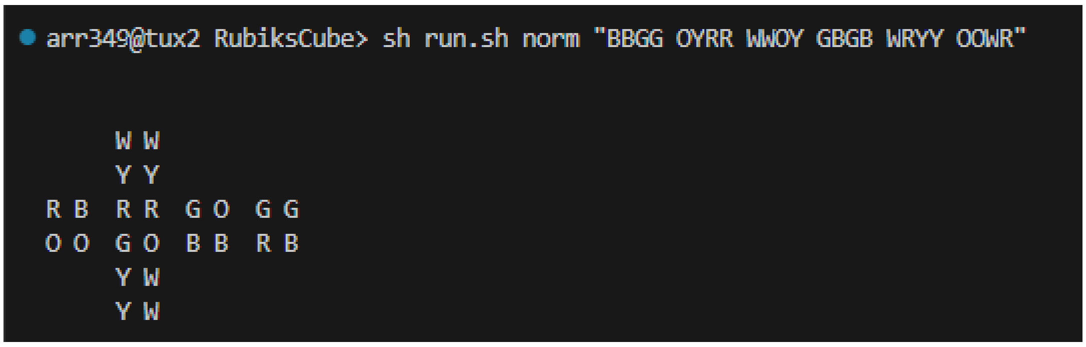
```
● arr349@tux2 RubiksCube> sh run.sh dls "L D' R' F R D'" 8
U U U' F' R' U R U'
     B W                              G B                              W G
     G W                              W W                              W B
O Y  R R  B B  O Y               R R  B B  O Y  O Y               B B  O Y  O Y  R R
W R  B Y  O O  W G               W R  B Y  O O  W G               W R  B Y  O O  W G
     Y G                              Y G                              Y G
     R G                              R G                              R G

     G B                              G B                              G W
     W W                              O O                              O O
R R  B B  O Y  O Y               R W  B Y  G Y  O Y               R W  B B  Y O  G Y
W R  B Y  O O  W G               W W  B B  Y O  W G               W W  B O  G Y  R G
     Y G                              R R                              R Y
     R G                              R G                              R B

     O G                              O O                              O O
     O W                              O O                              O O
B B  Y O  G Y  R W               B B  Y Y  G G  W W               W W  B B  Y Y  G G
W W  B O  G Y  R G               W W  B B  Y Y  G G               W W  B B  Y Y  G G
     R Y                              R R                              R R
     R B                              R R                              R R

390249
3.9417219161987305
```



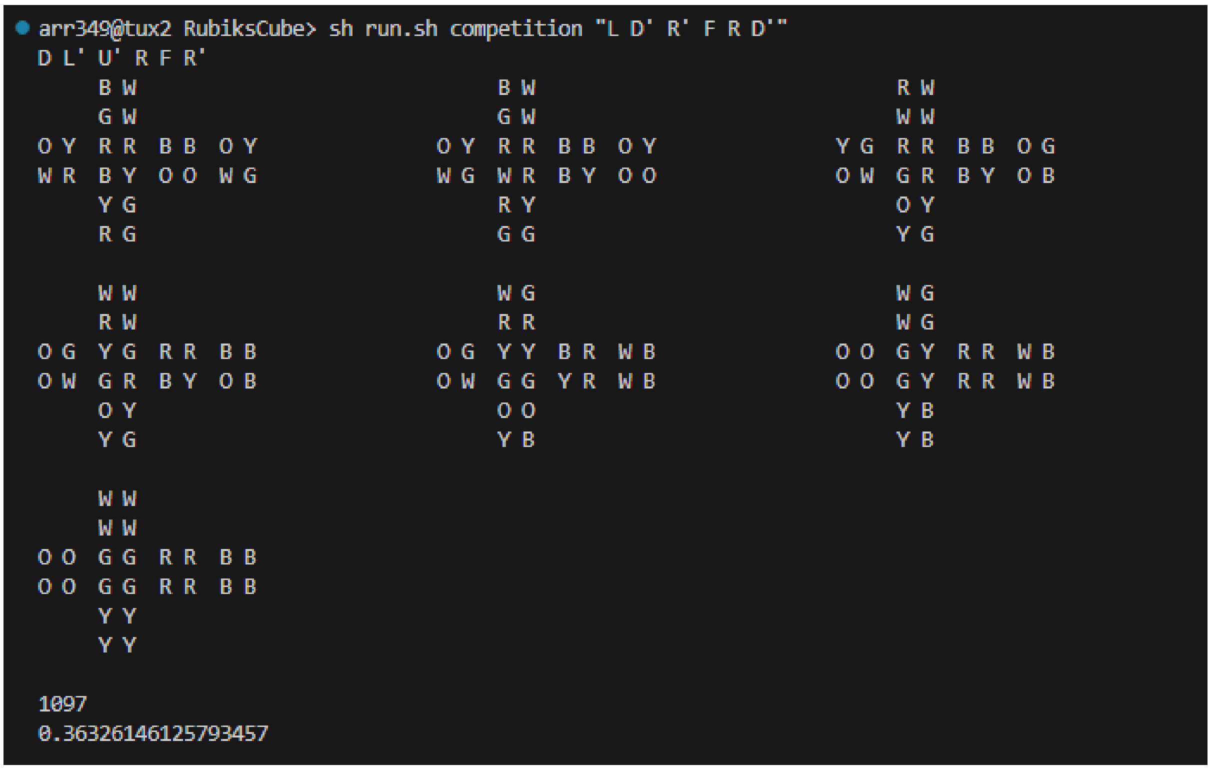
```
arr349@tux2 RubiksCube> sh run.sh ids "L D' R' F R D'" 20
Depth: 0 d: 1
Depth: 1 d: 13
Depth: 2 d: 157
Depth: 3 d: 1885
Depth: 4 d: 22621
Depth: 5 d: 271453
Depth: 6 d: 118794
IDS solution found at depth 6
U F' R' U R U'
     B W                              G B                              G B
     G W                              W W                              O O
O Y  R R  B B  O Y               R R  B B  O Y  O Y               R W  B Y  G Y  O Y
W R  B Y  O O  W G               W R  B Y  O O  W G               W W  B B  Y O  W G
     Y G                              Y G                              R R
     R G                              R G                              R G

     G W                              O G                              O O
     O O                              O W                              O O
R W  B B  Y O  G Y               B B  Y O  G Y  R W               B B  Y Y  G G  W W
W W  B O  G Y  R G               W W  B O  G Y  R G               W W  B B  Y Y  G G
     R Y                              R Y                              R R
     R B                              R B                              R R

     O O
     O O
W W  B B  Y Y  G G
W W  B B  Y Y  G G
     R R
     R R

Total iterations: 414924
4.7250237464904785
```



```
arr349@tux2 RubiksCube> sh run.sh astar "L D' R' F R D'"
D L' U' R F R'
     B W                              B W                              R W
     G W                              G W                              W W
O Y  R R  B B  O Y               O Y  R R  B B  O Y               Y G  R R  B B  O G
W R  B Y  O O  W G               W G  W R  B Y  O O               O W  G R  B Y  O B
     Y G                              R Y                              O Y
     R G                              G G                              Y G

     W W                              W G                              W G
     R W                              R R                              W G
O G  Y G  R R  B B               O G  Y Y  B R  W B               O O  G Y  R R  W B
O W  G R  B Y  O B               O W  G G  Y R  W B               O O  G Y  R R  W B
     O Y                              O O                              Y B
     Y G                              Y B                              Y B

     W W
     W W
O O  G G  R R  B B
O O  G G  R R  B B
     Y Y
     Y Y

1097
0.37050366401672363
```



```
● arr349@tux2 RubiksCube> sh run.sh norm "BBGG OYRR WWOY GBGB WRYY OOWR"


      W W
      Y Y
R B   R R   G O   G G
O O   G O   B B   R B
      Y W
      Y W
```



```
● arr349@tux2 RubiksCube> sh run.sh competition "L D' R' F R D'"
  D L' U' R F R'
       B W                               B W                               R W
       G W                               G W                               W W
  O Y  R R  B B  O Y                O Y  R R  B B  O Y                Y G  R R  B B  O G
  W R  B Y  O O  W G                W G  W R  B Y  O O                O W  G R  B Y  O B
       Y G                               R Y                               O Y
       R G                               G G                               Y G

       W W                               W G                               W G
       R W                               R R                               W G
  O G  Y G  R R  B B                O G  Y Y  B R  W B                O O  G Y  R R  W B
  O W  G R  B Y  O B                O W  G G  Y R  W B                O O  G Y  R R  W B
       O Y                               O O                               Y B
       Y G                               Y B                               Y B

       W W
       W W
  O O  G G  R R  B B
  O O  G G  R R  B B
       Y Y
       Y Y

  1097
  0.36326146125793457
```