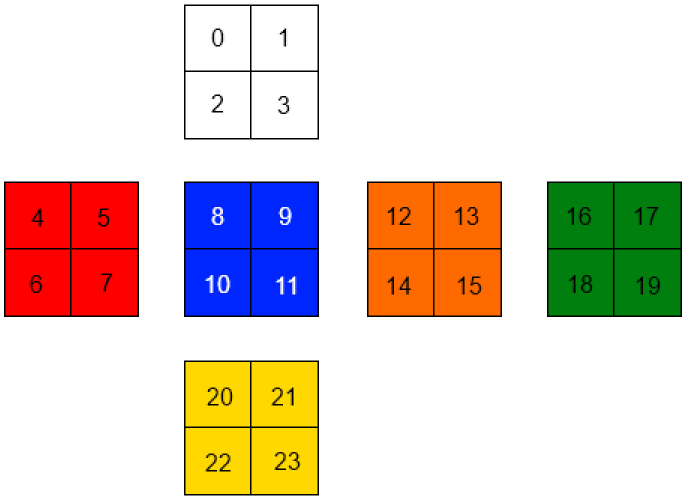
P1 Extra Credit Writeup

Code is run from outside of the package like so: `java P1.ECParser P1/exampleE1.txt`

For the Rubik's cube I used only one heuristic, unlike the original problem which used two for A*.

The heuristic used is an incorrect-face count. The faces of the center side are compared with one another. Two tiles are compared from the adjacent sides. The opposite side is not checked because its correctness is already known by process of elimination.



The cube is represented as a character array with indices matching the image above. Any orientation is correct, as long as white is opposite yellow, red is opposite orange, and blue is opposite green.

Code Correctness

To verify correctness, a number of examples are used. Each example has a corresponding `.txt` file.

Be sure to use `ECParser` and not `Parser`!

Example 1 – Goal state base case

```
java P1.ECParser P1/exampleE1.txt
```

Example 1 shows that both algorithms successfully halt when the goal state is the intial state.

Example 2 – Low entropy

```
java P1.ECParser P1/exampleE2.txt
```

Example 2 shows that both algorithms successfully find the goal state from a state requiring only a small few moves. In this case, the LBS is different A* search. Additionally, we can see that the `randomizeState` command always generates the same state after five moves.

Example 3 – High entropy

```
java P1.ECParser P1/exampleE3.txt
```

Example 3 shows that it is not feasible to solve a high entropy cube using BFS techniques. Both A* and LBS fail.