

BIE-ZUM Riku Kawasaki

Topic: 3x3x3 Rubik's cube solver

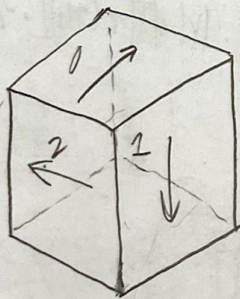
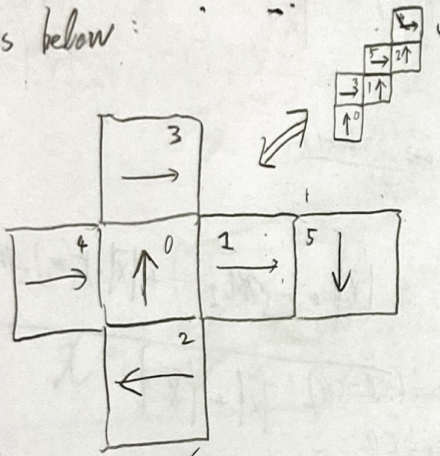
Used Algorithm: A* Algorithm

Data Structure:

• Cube

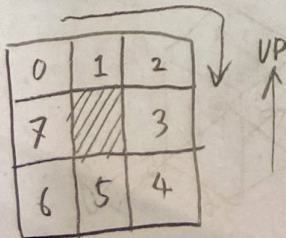
consists of 6 faces indexed

as below:



• Face

consists of 8 panels with ID clockwise



State Space

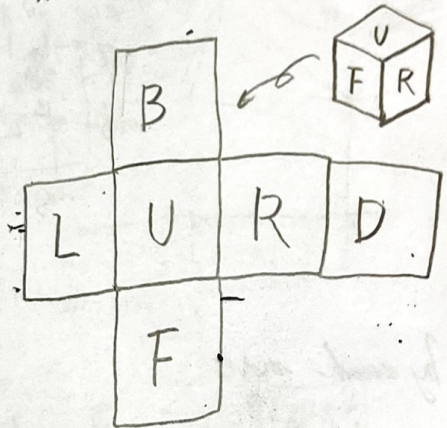
State $S = \{ \text{All the possible Cubes} \}$

Action $A = \{ U, D, R, L, F, B, U', D', R', L', F', B' \}$

for $X \in A$,

X --- move X clockwise

X' --- move X anti-clockwise.



Initial State I = shuffled Cube

Goal state G = Cube where all the panels are at the right position.

cost $C = 1.0$ (per each Action)

Heuristic Function $h(s)$

Let B_s be a set of blocks of the cube at the state s then,

$$h(s) = \sum_{b \in B_s} (\text{the number of the shortest moves for } b \text{ to reach the desired position})$$

The value is pre-calculated by BFS for any possible combinations of block positions.

Thus, the evaluation function :

$$f(s) = g(s) + h(s)$$

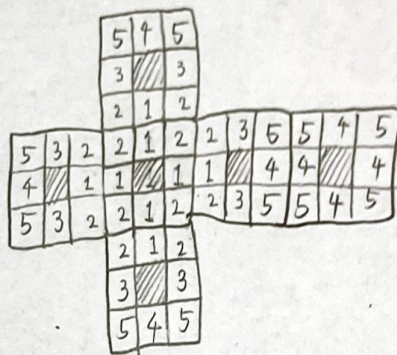
($g(s) = 1.0$)

Ingenuity

Even with the heuristic function, the pruning space was too big to reach some good states.

To avoid getting lost on the way to the good states, I set 17 subgoals with mask.

My program try to complete the panels in this order :
(from 1~5)



By this, the program can put back 28~42 panels out of 48 to the right position.

Summary

My program couldn't complete solving the puzzle. (On avg. around 70% of blocks could be placed back correctly.) That indicates more precise heuristics is necessary.