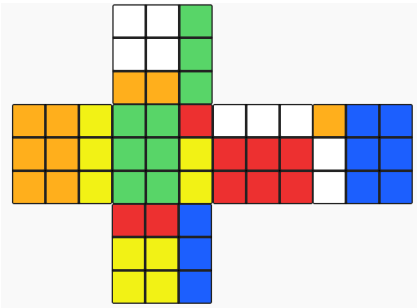
# Cube-Result

## 三阶

六步：1711s（搜索了三百万个状态）
五步：18s
四步：2s
三步：50ms
两步：20ms

状态1：



最优需要两步
对应：

```
    eq init = (
    (f[face(1)]: C4 C4 C4 C1 C1 C5 C1 C1 C5)
    (f[face(2)]: C2 C2 C2 C2 C2 C2 C5 C1 C1)
    (f[face(3)]: C1 C3 C3 C1 C3 C3 C1 C3 C3)
    (f[face(4)]: C3 C6 C6 C4 C4 C4 C4 C4 C4)
    (f[face(5)]: C6 C6 C6 C5 C5 C5 C5 C5 C5)
    (f[face(6)]: C2 C2 C2 C3 C6 C6 C3 C6 C6)
    ) .
```

```
Maude> search [1, 2]  init =>* S:State such that success(S:State) .
search [1, 2] in CUBE : init =>* S:State such that success(S:State) = true .

Solution 1 (state 136)
states: 137  rewrites: 216183 in 20ms cpu (20ms real) (10809150 rewrites/second)
S:State --> (f[face(1)]: C1 C1 C1 C1 C1 C1 C1 C1 C1) (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2) (f[face(3)]: C3 C3 C3
    C3 C3 C3 C3 C3 C3) (f[face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4) (f[face(5)]: C5 C5 C5 C5 C5 C5 C5 C5 C5) f[face(6)]:
    C6 C6 C6 C6 C6 C6 C6 C6 C6

# path
Maude> show path 136 .
state 0, State: (f[face(1)]: C4 C4 C4 C1 C1 C5 C1 C1 C5) (f[face(2)]: C2 C2 C2 C2 C2 C2 C5 C1 C1) (f[face(3)]: C1 C3
    C3 C1 C3 C3 C1 C3 C3) (f[face(4)]: C3 C6 C6 C4 C4 C4 C4 C4 C4) (f[face(5)]: C6 C6 C6 C5 C5 C5 C5 C5 C5) f[face(
    6)]: C2 C2 C2 C3 C6 C6 C3 C6 C6
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL)
    f[face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR,
    adjacent(N, RIGHT))) (f[face(RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[
    face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]:
    SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT ==
    rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 9, State: (f[face(1)]: C1 C1 C5 C1 C1 C5 C1 C1 C5) (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2) (f[face(3)]: C3 C3
    C3 C3 C3 C3 C1 C1 C1) (f[face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4) (f[face(5)]: C6 C6 C6 C5 C5 C5 C5 C5 C5) f[face(
    6)]: C3 C6 C6 C3 C6 C6 C3 C6 C6
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL)
    f[face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR,
    adjacent(N, RIGHT))) (f[face(RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[
    face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]:
    SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT ==
    rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 136, State: (f[face(1)]: C1 C1 C1 C1 C1 C1 C1 C1 C1) (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2) (f[face(3)]: C3
    C3 C3 C3 C3 C3 C3 C3 C3) (f[face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4) (f[face(5)]: C5 C5 C5 C5 C5 C5 C5 C5 C5) f[
    face(6)]: C6 C6 C6 C6 C6 C6 C6 C6 C6
```

状态2：本就是复原状态

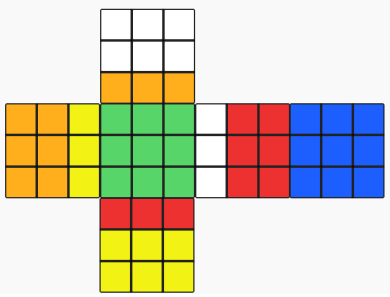
```
    eq init = (
    (f[face(1)]: C1 C1 C1 C1 C1 C1 C1 C1 C1)
    (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2)
```

```
    (f[face(3)]: C3 C3 C3 C3 C3 C3 C3 C3 C3)
    (f[face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4)
    (f[face(5)]: C5 C5 C5 C5 C5 C5 C5 C5 C5)
    (f[face(6)]: C6 C6 C6 C6 C6 C6 C6 C6 C6)
    ) .
```

结果：

```
Maude> search [1, 3]  init =>* S:State such that success(S:State) .
search [1, 3] in CUBE : init =>* S:State such
    that success(S:State) = true .

Solution 1 (state 0)
states: 1  rewrites: 13 in 0ms cpu (0ms real) (~
    rewrites/second)
S:State --> (f[face(1)]: C1 C1 C1 C1 C1 C1 C1 C1
    C1) (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2)
    (f[face(3)]: C3 C3 C3 C3 C3 C3 C3 C3 C3) (f[
    face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4) (f[
    face(5)]: C5 C5 C5 C5 C5 C5 C5 C5 C5) f[
    face(6)]: C6 C6 C6 C6 C6 C6 C6 C6 C6
```

状态3：

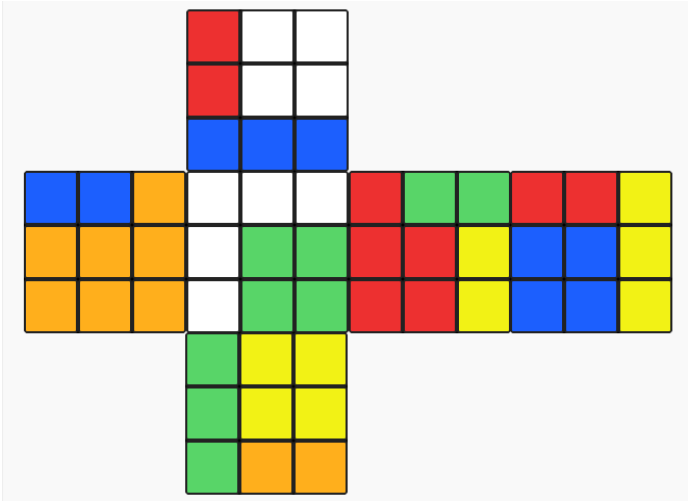


最优需要一步：

```
    eq init = (
    (f[face(1)]: C1 C1 C5 C1 C1 C5 C1 C1 C5)
    (f[face(2)]: C2 C2 C2 C2 C2 C2 C2 C2 C2)
    (f[face(3)]: C3 C3 C3 C3 C3 C3 C1 C1 C1)
    (f[face(4)]: C4 C4 C4 C4 C4 C4 C4 C4 C4)
    (f[face(5)]: C6 C6 C6 C5 C5 C5 C5 C5 C5)
    (f[face(6)]: C3 C6 C6 C3 C6 C6 C3 C6 C6)
    ) .
```

结果：

```
Maude> search [1, 3]  init =>* S:State such that success(S:State) .
search [1, 3] in CUBE : init =>* S:State such
    that success(S:State) = true .

Solution 1 (state 10)
states: 11  rewrites: 13035 in 10ms cpu (1ms
    real) (1303500 rewrites/second)
S:State --> (f[face(1)]: W W W W W W W W W) (f[
    face(2)]: B B B B B B B B B) (f[face(3)]: R
    R R R R R R R R) (f[face(4)]: G G G G G G G
    G G) (f[face(5)]: O O O O O O O O O) f[face(
    6)]: Y Y Y Y Y Y Y Y Y
```

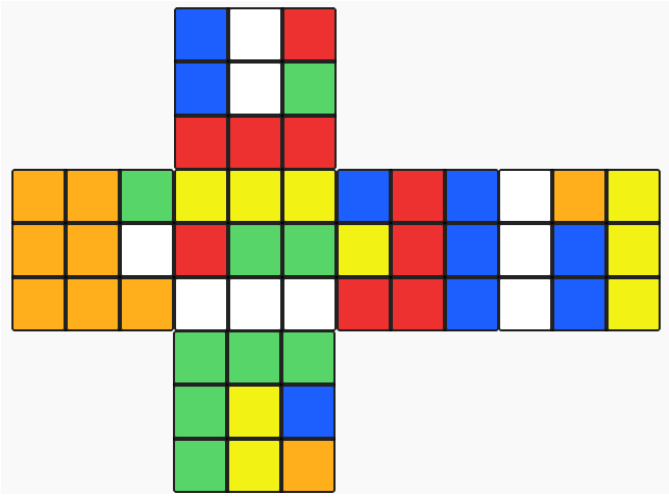


最优需要三步：

```
    eq init = (
    (f[face(1)]: W W B W W B R R B)
    (f[face(2)]: Y Y Y R B B R B B)
    (f[face(3)]: G Y Y G R R R R R)
    (f[face(4)]: W G G W G G W W W)
    (f[face(5)]: O O O B O O B O O)
    (f[face(6)]: Y Y O Y Y O G G G)
    ) .
```

结果：

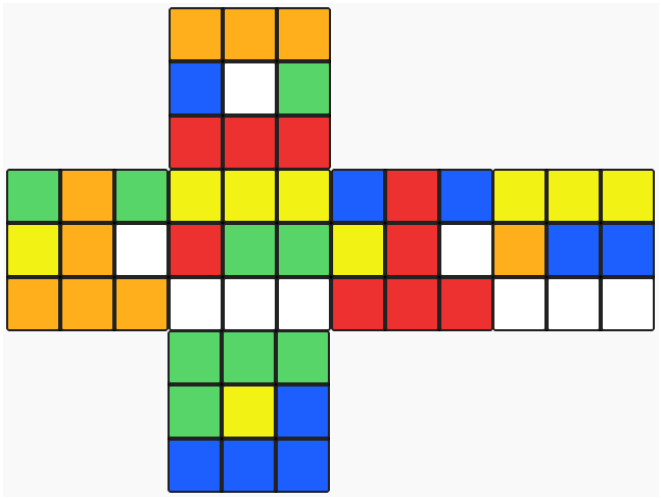
```
Solution 1 (state 349)
states: 350  rewrites: 597097 in 50ms cpu (56ms real) (11941940 rewrites/second)
S:State --> (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G G G G G G G G G)
(f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
# path
Maude> show path 349 .
state 0, State: (f[face(1)]: W W B W W B R R B) (f[face(2)]: Y Y Y R B B R B B) (f[face(3)]: G Y Y G R R R R R) (f[face(4)]: W G G W G G W W W)
(f[face(5)]: O O O B O O B O O) f[face(6)]: Y Y O Y Y O G G G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 1, State: (f[face(1)]: R W W R W W B B B) (f[face(2)]: O Y Y B B B B B B) (f[face(3)]: Y Y Y R R R R R R) (f[face(4)]: G G G G G G R W W)
(f[face(5)]: W O O W O O W O O) f[face(6)]: Y Y O Y Y O G G G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 26, State: (f[face(1)]: R W W R W W R W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: Y Y Y R R R R R R) (f[face(4)]: G G G G G G G G G)
(f[face(5)]: O O O O O O W W W) f[face(6)]: Y Y O Y Y O Y Y O
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 349, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G G G G G G G G G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
```



最优需要四步

```
    eq init = (
    (f[face(1)]: R G R W W R B B R)
    (f[face(2)]: Y Y Y O B B W W W)
    (f[face(3)]: B B B R R R B Y R)
    (f[face(4)]: Y G W Y G W Y R W)
    (f[face(5)]: G W O O O O O O O)
    (f[face(6)]: G B O G Y Y G G G)
    ) .
```

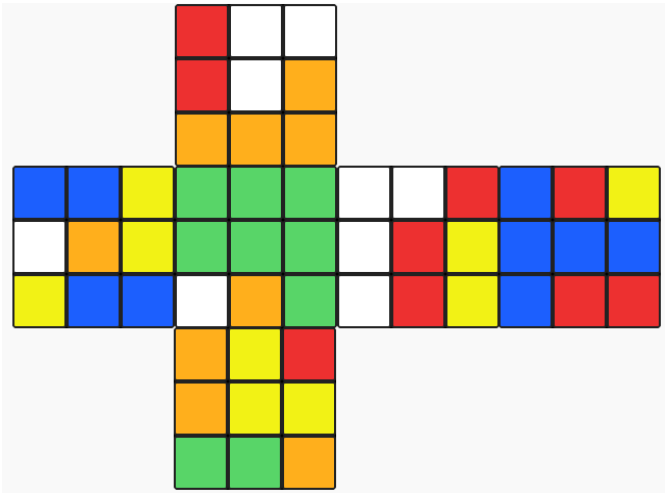
```
Maude> search [1, 4]  init =>* S:State such that success(S:State) .
search [1, 4] in CUBE : init =>* S:State such that success(S:State) = true .

Solution 1 (state 11187)
states: 11188  rewrites: 20104786 in 2160ms cpu (2161ms real) (9307771 rewrites/second)
S:State --> (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G
    G G G G G G G G G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
# path
Maude> show path 11187 .
state 0, State: (f[face(1)]: R G R W W R B B R) (f[face(2)]: Y Y Y O B B W W W) (f[face(3)]: B B B R R R B Y R) (f[face(4)]: Y G W Y G W Y R W)
(f[face(5)]: G W O O O O O O O) f[face(6)]: G B O G Y Y G G G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 4, State: (f[face(1)]: R G G W W W B B O) (f[face(2)]: Y Y Y O B B W W W) (f[face(3)]: B B B R R R R R R) (f[face(4)]: Y Y Y R G G W W W)
(f[face(5)]: G G G O O O O O O) f[face(6)]: B B O Y Y Y R G G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 69, State: (f[face(1)]: W W W W W W B B O) (f[face(2)]: Y Y Y O B B O B B) (f[face(3)]: B R R B R R B R R) (f[face(4)]: R G G R G G W W W)
(f[face(5)]: G G G O O O O O O) f[face(6)]: Y Y Y Y Y Y R G G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 891, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: O B B O B B O B B) (f[face(3)]: B R R B R R B R R) (f[face(4)]: R G G R G G R G
G) (f[face(5)]: G O O G O O G O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 11187, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G G G G G G G G
G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
```



最优需要五步

```
    eq init = (
    (f[face(1)]: O G R O W R O B R)
    (f[face(2)]: Y B W Y B W Y O W)
    (f[face(3)]: B W R R R R B Y R)
    (f[face(4)]: Y G W Y G W Y R W)
    (f[face(5)]: G W O O O O G Y O)
    (f[face(6)]: G B B G Y B G G B)
    ) .
```

```
Maude> search [1, 5]  init =>* S:State such that success(S:State) .
search [1, 5] in CUBE : init =>* S:State such that success(S:State) = true .

Solution 1 (state 72166)
states: 72167  rewrites: 130862578 in 18620ms cpu (18611ms real) (7028065 rewrites/second)
S:State --> (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G
    G G G G G G G G G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
# path
Maude> show path 72166 .
state 0, State: (f[face(1)]: O G R O W R O B R) (f[face(2)]: Y B W Y B W Y O W) (f[face(3)]: B W R R R R B Y R) (f[face(4)]: Y G W Y G W Y R W)
(f[face(5)]: G W O O O O G Y O) f[face(6)]: G B B G Y B G G B
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 2, State: (f[face(1)]: R G R W W R B B R) (f[face(2)]: Y Y Y O B B W W W) (f[face(3)]: B B B R R R B Y R) (f[face(4)]: Y G W Y G W Y R W)
(f[face(5)]: G W O O O O O O O) f[face(6)]: G B O G Y Y G G G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 35, State: (f[face(1)]: R G G W W W B B O) (f[face(2)]: Y Y Y O B B W W W) (f[face(3)]: B B B R R R R R R) (f[face(4)]: Y Y Y R G G W W W)
(f[face(5)]: G G G O O O O O O) f[face(6)]: B B O Y Y Y R G G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 460, State: (f[face(1)]: W W W W W W B B O) (f[face(2)]: Y Y Y O B B O B B) (f[face(3)]: B R R B R R B R R) (f[face(4)]: R G G R G G W W
W) (f[face(5)]: G G G O O O O O O) f[face(6)]: Y Y Y Y Y Y R G G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 5802, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: O B B O B B O B B) (f[face(3)]: B R R B R R B R R) (f[face(4)]: R G G R G G R G
G) (f[face(5)]: G O O G O O G O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 72166, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G G G G G G G G
G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
```



最优需要六步：

```
    eq init = (
    (f[face(1)]: W O O W W O R R O)
    (f[face(2)]: Y B R R B R B B B)
    (f[face(3)]: R Y Y W R R W W W)
    (f[face(4)]: G G G G G O G G W)
    (f[face(5)]: Y Y B B O B B W Y)
    (f[face(6)]: R Y O Y Y G O O G)
    ) .
```

结果：

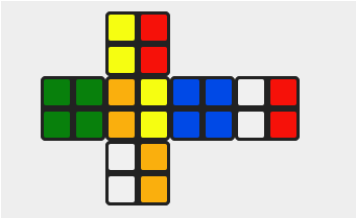
```
Maude> search [1, 6] init =>* S:State such that success(S:State) .
search [1, 6] in CUBE : init =>* S:State such that success(S:State) = true .

Solution 1 (state 3327424)
states: 3327425  rewrites: 61111546645 in 1711600ms cpu (1711592ms real) (3570433 rewrites/second)
S:State --> (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(
    4)]: G G G G G G G G G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
# path
Maude> show path 3327424 .
state 0, State: (f[face(1)]: W O O W W O R R O) (f[face(2)]: Y B R R B R B B B) (f[face(3)]: R Y Y W R R W W W) (f[face(4)]: G G G G G O G G W)
(f[face(5)]: Y Y B B O B B W Y) f[face(6)]: R Y O Y Y G O O G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 9, State: (f[face(1)]: B B B W W O R R O) (f[face(2)]: Y B R R B R O Y R) (f[face(3)]: Y R W Y R W R W W) (f[face(4)]: W O O G G O G G W)
(f[face(5)]: Y Y B B O B B W Y) f[face(6)]: G G G Y Y G O O G
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 138, State: (f[face(1)]: B B B W W O R R O) (f[face(2)]: Y B B R B B O Y Y) (f[face(3)]: Y R R Y R R R W R) (f[face(4)]: W O W G G W G G
W) (f[face(5)]: Y Y O B O O B W W) f[face(6)]: G G G G Y O G Y O
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 1751, State: (f[face(1)]: B B R W W W R R R) (f[face(2)]: Y B B R B B O Y Y) (f[face(3)]: Y R R Y R R G G G) (f[face(4)]: W W W O G G W G
G) (f[face(5)]: B O O B O O B W W) f[face(6)]: O G G Y Y O Y Y O
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 21903, State: (f[face(1)]: W W W W W W R R R) (f[face(2)]: Y B B R B B R B B) (f[face(3)]: G Y Y G R R G R R) (f[face(4)]: O G G O G G W G
G) (f[face(5)]: B O O B O O B W W) f[face(6)]: Y Y O Y Y O Y Y O
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRL, adjacent(N, LEFT))) (f[face(RIGHT)]:
    SetThroughTriple(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM), CRR,
adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 270988, State: (f[face(1)]: R W W R W W R W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: Y Y Y R R R R R R) (f[face(4)]: G G G G G G G
```

```
G G) (f[face(5)]: O O O O O O W W W) f[face(6)]: Y Y O Y Y O Y Y O
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(RIGHT)]: CRR =>
(f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughTriple(CRT, adjacent(N, TOP), CRR, adjacent(N, RIGHT))) (f[face(
    RIGHT)]: SetThroughTriple(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]: SetThroughTriple(CRB, adjacent(N, BOTTOM),
CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughTriple(CRL, adjacent(N, LEFT), CRT, adjacent(N, TOP)) if ((LEFT ==
    leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 3327424, State: (f[face(1)]: W W W W W W W W W) (f[face(2)]: B B B B B B B B B) (f[face(3)]: R R R R R R R R R) (f[face(4)]: G G G G G G G
G G) (f[face(5)]: O O O O O O O O O) f[face(6)]: Y Y Y Y Y Y Y Y Y
```

## 二阶

七步需要：4038s
六步需要：123s
五步需要：10s
四步需要：1s

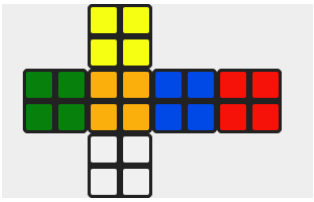
search [1, 3] init =>* S:State such that success(S:State) .



```
    eq init = (
    (f[face(1)]: C2 C2 C1 C1)
    (f[face(2)]: C2 C2 C6 C6)
    (f[face(3)]: C3 C3 C3 C3)
    (f[face(4)]: C1 C1 C4 C4)
    (f[face(5)]: C5 C5 C5 C5)
    (f[face(6)]: C4 C4 C6 C6)
    ) .
```

结果：

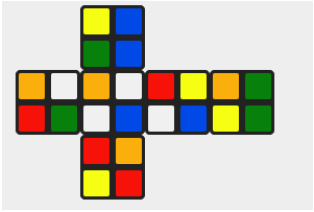
```
Maude> search [1, 100]  init =>* S:State such that success(S:State) .
search [1, 100] in CUBE-2 : init =>* S:State such that success(S:State) = true .

Solution 1 (state 3)
states: 4  rewrites: 3119 in 0ms cpu (0ms real) (~ rewrites/second)
S:State --> (f[face(1)]: C1 C1 C1 C1) (f[face(2)]: C2 C2 C2 C2) (f[face(3)]: C3 C3 C3 C3) (f[face(4)]: C4 C4 C4
    C4) (f[face(5)]: C5 C5 C5 C5) f[face(6)]: C6 C6 C6 C6
```



```
    eq init = (
    (f[face(1)]: C1 C1 C1 C1)
    (f[face(2)]: C2 C2 C2 C2)
    (f[face(3)]: C3 C3 C3 C3)
    (f[face(4)]: C4 C4 C4 C4)
    (f[face(5)]: C5 C5 C5 C5)
    (f[face(6)]: C6 C6 C6 C6)
    ) .
```

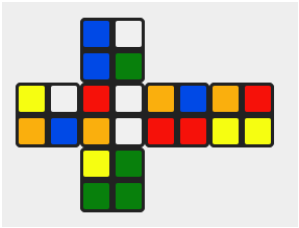
结果：

```
Maude> search [1, 100]  init =>* S:State such that success(S:State) .
search [1, 100] in CUBE-2 : init =>* S:State such that success(S:State) = true .

Solution 1 (state 0)
states: 1  rewrites: 13 in 0ms cpu (0ms real) (~ rewrites/second)
S:State --> (f[face(1)]: C1 C1 C1 C1) (f[face(2)]: C2 C2 C2 C2) (f[face(3)]: C3 C3 C3 C3) (f[face(4)]: C4 C4 C4
    C4) (f[face(5)]: C5 C5 C5 C5) f[face(6)]: C6 C6 C6 C6
```



```
    eq init = (
    (f[face(1)]: B B Y G)
    (f[face(2)]: G G O Y)
    (f[face(3)]: Y B R W)
    (f[face(4)]: W B O W)
    (f[face(5)]: W G O R)
    (f[face(6)]: O R R Y)
    ) .
```

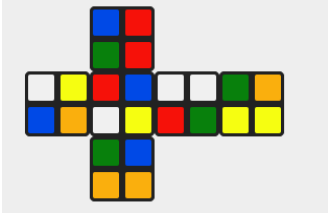
结果：最优需要9步



最优需要7步

```
    eq init = (
    (f[face(1)]: W G B B)
    (f[face(2)]: R Y O Y)
    (f[face(3)]: B R O R)
    (f[face(4)]: W W R O)
    (f[face(5)]: W B Y O)
    (f[face(6)]: G G Y G)
    ) .
```

结果：

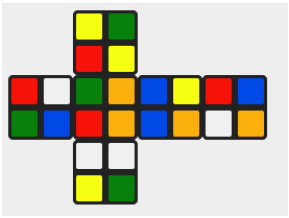


```
    eq init = (
    (f[face(1)]: R R B G)
    (f[face(2)]: O Y G Y)
    (f[face(3)]: W G W R)
    (f[face(4)]: B Y R W)
    (f[face(5)]: Y O W B)
    (f[face(6)]: B O G O)
    ) .
```

最优需要四步：
结果：

```
Maude> search [1, 4]  init =>* S:State such that success(S:State) .
search [1, 4] in CUBE-2 : init =>* S:State such that success(S:State) = true .

Solution 1 (state 5050)
states: 5051  rewrites: 110793347 in 1060ms cpu (1054ms real) (10452214 rewrites/second)
S:State --> (f[face(1)]: B B B B) (f[face(2)]: Y Y Y Y) (f[face(3)]: R R R R) (f[face(4)]: W W W W) (f[face(5)]: O O O O)
    f[face(6)]: G G G G
```

```
# 路径

Maude> show path 5050 .
state 0, State: (f[face(1)]: R R B G) (f[face(2)]: O Y G Y) (f[face(3)]: W G W R) (f[face(4)]: B Y R W) (f[face(5)]: Y O W
    B) f[face(6)]: B O G O
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[
    face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRR, adjacent(
    N, RIGHT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(
    N)) and TOP == topNeighbor(N) = true . ]===>
state 7, State: (f[face(1)]: R G R B) (f[face(2)]: W Y W Y) (f[face(3)]: B G R R) (f[face(4)]: Y Y W W) (f[face(5)]: O O G
    B) f[face(6)]: B O G O
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[
    face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRR, adjacent(
    N, RIGHT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(
    N)) and TOP == topNeighbor(N) = true . ]===>
state 80, State: (f[face(1)]: G G B B) (f[face(2)]: Y Y W W) (f[face(3)]: R R R R) (f[face(4)]: Y Y W W) (f[face(5)]: O O O
    O) f[face(6)]: B B G G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(
    RIGHT)]: CRR => (f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRL, adjacent(N,
    LEFT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRR, adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM ==
    bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 682, State: (f[face(1)]: Y Y B B) (f[face(2)]: Y Y G G) (f[face(3)]: R R R R) (f[face(4)]: B B W W) (f[face(5)]: O O
    O O) f[face(6)]: W W G G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(
    RIGHT)]: CRR => (f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRL, adjacent(N,
    LEFT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRR, adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM ==
    bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 5050, State: (f[face(1)]: B B B B) (f[face(2)]: Y Y Y Y) (f[face(3)]: R R R R) (f[face(4)]: W W W W) (f[face(5)]: O O
    O O) f[face(6)]: G G G G
```



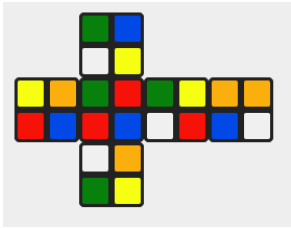
最优需要五步：

```
    eq init = (
    (f[face(1)]: G Y Y R)
    (f[face(2)]: B O R W)
    (f[face(3)]: Y O B B)
    (f[face(4)]: O O G R)
    (f[face(5)]: W B R G)
    (f[face(6)]: W G W Y)
    ) .
```

结果：

```
Maude> search [1, 5]  init =>* S:State such that success(S:State) .
search [1, 5] in CUBE-2 : init =>* S:State such that success(S:State) = true .

Solution 1 (state 40931)
states: 40932  rewrites: 1025417773 in 10840ms cpu (10844ms real) (9459573 rewrites/second)
S:State --> (f[face(1)]: Y Y Y Y) (f[face(2)]: O O O O) (f[face(3)]: G G G G) (f[face(4)]: R R R R) (f[face(5)]: B B B B)
    f[face(6)]: W W W W
# 路径
Maude> show path 40931 .
state 0, State: (f[face(1)]: G Y Y R) (f[face(2)]: B O R W) (f[face(3)]: Y O B B) (f[face(4)]: O O G R) (f[face(5)]: W B R
    G) f[face(6)]: W G W Y
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[
    face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRR, adjacent(
    N, RIGHT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(
    N)) and TOP == topNeighbor(N) = true . ]===>
state 9, State: (f[face(1)]: W R Y R) (f[face(2)]: B O G W) (f[face(3)]: O B Y B) (f[face(4)]: G Y G R) (f[face(5)]: W B R
```

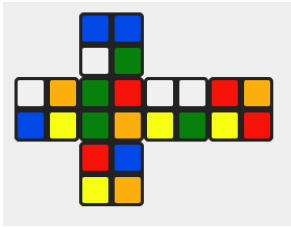
```
    G) f[face(6)]: O O W Y
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[
    face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRR, adjacent(
    N, RIGHT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(
    N)) and TOP == topNeighbor(N) = true . ]===>
state 98, State: (f[face(1)]: R R W Y) (f[face(2)]: O O Y W) (f[face(3)]: G B G B) (f[face(4)]: W Y R R) (f[face(5)]: B B G
    G) f[face(6)]: O O W Y
===[ crl [RotateCounterClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[
    face(RIGHT)]: CRR => (f[face(N)]: rotateCCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRR, adjacent(
    N, RIGHT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRB, adjacent(N, BOTTOM))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRL, adjacent(N, LEFT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRT, adjacent(N, TOP)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM == bottomNeighbor(
    N)) and TOP == topNeighbor(N) = true . ]===>
state 846, State: (f[face(1)]: W Y W Y) (f[face(2)]: O O O O) (f[face(3)]: B B G G) (f[face(4)]: R R R R) (f[face(5)]: B B
    G G) f[face(6)]: W Y W Y
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(
    RIGHT)]: CRR => (f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRL, adjacent(N,
    LEFT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRR, adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM ==
    bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 6318, State: (f[face(1)]: B Y B Y) (f[face(2)]: O O O O) (f[face(3)]: Y Y G G) (f[face(4)]: R R R R) (f[face(5)]: B B
    W W) f[face(6)]: W G W G
===[ crl [RotateClockWise] : (f[face(N)]: CR) (f[face(TOP)]: CRT) (f[face(BOTTOM)]: CRB) (f[face(LEFT)]: CRL) f[face(
    RIGHT)]: CRR => (f[face(N)]: rotateCW(CR)) (f[face(TOP)]: SetThroughDouble(CRT, adjacent(N, TOP), CRL, adjacent(N,
    LEFT))) (f[face(RIGHT)]: SetThroughDouble(CRR, adjacent(N, RIGHT), CRT, adjacent(N, TOP))) (f[face(BOTTOM)]:
    SetThroughDouble(CRB, adjacent(N, BOTTOM), CRR, adjacent(N, RIGHT))) f[face(LEFT)]: SetThroughDouble(CRL, adjacent(N,
    LEFT), CRB, adjacent(N, BOTTOM)) if ((LEFT == leftNeighbor(N) and RIGHT == rightNeighbor(N)) and BOTTOM ==
    bottomNeighbor(N)) and TOP == topNeighbor(N) = true . ]===>
state 40931, State: (f[face(1)]: Y Y Y Y) (f[face(2)]: O O O O) (f[face(3)]: G G G G) (f[face(4)]: R R R R) (f[face(5)]: B
    B B B) f[face(6)]: W W W W
```



最优需要七步：

```
    eq init = (
    (f[face(1)]: B Y G W)
    (f[face(2)]: O W O B)
    (f[face(3)]: Y R G W)
    (f[face(4)]: R B G R)
    (f[face(5)]: O B Y R)
    (f[face(6)]: O Y W G)
    ) .
```

可以解，求解时间前面给出了，结果忘记贴了。



最优需要13步，无法解，时间太长

```
    eq init = (
    (f[face(1)]: B G B W)
    (f[face(2)]: O R R Y)
    (f[face(3)]: W G W Y)
    (f[face(4)]: R O G G)
    (f[face(5)]: O Y W B)
```

```
    (f[face(6)]: B O R Y)
    ) .
```