

RubikCube 2x2 解き方 フローチャート
TadaMatz ver. 1/2

- 前提とかメモとか
- ・回転の種類を4方向{①, ②, ③, ④}×3角度{90, -90, 180}=12種類として番号をふる. 数(列)と動作を対応付ける.
 - ・幅優先探索っぽいことをする. 数列の列挙→枝切り→数列内1回転ごとにjudgeを行う
 - ・IREG=Initial Register, TREG=Temp Register, CREG=Count up Register, DEPTH = 最大回転動作数
 - ・同じ方向は連続しないから枝切りできる. (例: ①→①は続かない)

Cubeの表現方法について

- ・IREGとTREGはそれぞれレジスタ5個セットで24マスを表す. 下の図に示す番号が何ビット目かとcubeの位置との対応関係になっている. 5個のレジスタを縦に読んで同じ番目のビットをすべて足し合わせて6色を表現している. 例えば, 初めの4セル分を見てみると

IREG0 = 0011
IERG1 = 0110
IREG2 = 1110
IREG3 = 1111
IREG4 = 0111

とするとそれぞれの同じ番目のビットを足すと順に 2 4 5 3となる.
そのため下の図でL2には0bit目の2, L1には1bit目の4, L4には2bit目の5, L3には3bit目の3が対応する色になっているということになる.

↓

③

↓

④

16

T1

17

T2

18

T3

19

T4

④

↑

③

↑

0

L2

1

L1

4

F1

5

F2

8

R1

9

R2

12

B2

13

B1

2

L4

3

L3

6

F3

7

F4

10

R3

11

R4

14

B4

15

B3

20

B3

21

B4

22

B1

23

B2

③

↓

④

↓

→①

→②

モジュール

Rotate(register, number)

Judge(register)

レジスタ

InitialReg0

InitialReg1

InitialReg2

InitialReg3

InitialReg4

TempReg0

TempReg1

TempReg2

TempReg3

TempReg4

CountUpReg0

CountUpReg1

CountUpReg2

CountUpReg3

CountUpReg4

CountUpReg5

CountUpReg6

Itr1

Itr2

Result

出題された値を保存する. レイヤを5個用意するこ

とで6色を保存する

算出した数列に対して操作するとき用のレジスタ

列挙のための数列を保存するレジスタ. 数がある深さを表す. より深い探索の時はより多くする.

回転用いてれた

Judge結果保存

P=90度回転, N=-90度回転, T=180度回転

| 新しい位置 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 ①P | 12 | 13 | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 | 10 | 11 | 8 | 9 | 14 | 15 | 17 | 19 | 16 | 18 | 20 | 21 | 22 | 23 |
| 1 ①N | 4 | 5 | 2 | 3 | 8 | 9 | 6 | 7 | 12 | 13 | 10 | 11 | 0 | 1 | 14 | 15 | 18 | 16 | 19 | 17 | 20 | 21 | 22 | 23 |
| 2 ①T | 8 | 9 | 2 | 3 | 12 | 13 | 6 | 7 | 0 | 1 | 10 | 11 | 4 | 5 | 14 | 15 | 19 | 18 | 17 | 16 | 20 | 21 | 22 | 23 |
| 3 ②P | 0 | 1 | 14 | 15 | 4 | 5 | 2 | 3 | 8 | 9 | 6 | 7 | 12 | 13 | 10 | 11 | 16 | 17 | 18 | 19 | 22 | 20 | 23 | 21 |
| 4 ②N | 0 | 1 | 6 | 7 | 4 | 5 | 10 | 11 | 8 | 9 | 14 | 15 | 12 | 13 | 2 | 3 | 16 | 17 | 18 | 19 | 21 | 23 | 20 | 22 |
| 5 ②T | 0 | 1 | 10 | 11 | 4 | 5 | 14 | 15 | 8 | 9 | 2 | 3 | 12 | 13 | 6 | 7 | 16 | 17 | 18 | 19 | 23 | 22 | 21 | 20 |
| 6 ③P | 2 | 0 | 3 | 1 | 16 | 5 | 18 | 7 | 8 | 9 | 10 | 11 | 12 | 22 | 14 | 20 | 15 | 17 | 13 | 19 | 4 | 21 | 6 | 23 |
| 7 ③N | 1 | 3 | 0 | 2 | 20 | 5 | 22 | 7 | 8 | 9 | 10 | 11 | 12 | 18 | 14 | 16 | 4 | 17 | 6 | 19 | 15 | 21 | 13 | 23 |
| 8 ③T | 3 | 2 | 1 | 0 | 15 | 5 | 13 | 7 | 8 | 9 | 10 | 11 | 12 | 6 | 14 | 4 | 20 | 17 | 22 | 19 | 16 | 21 | 18 | 23 |
| 9 ④P | 0 | 1 | 2 | 3 | 4 | 17 | 6 | 19 | 9 | 11 | 8 | 10 | 23 | 13 | 21 | 15 | 16 | 14 | 18 | 12 | 20 | 5 | 22 | 7 |
| 10 ④N | 0 | 1 | 2 | 3 | 4 | 21 | 6 | 23 | 10 | 8 | 11 | 9 | 19 | 13 | 17 | 15 | 16 | 5 | 18 | 7 | 20 | 14 | 22 | 12 |
| 11 ④T | 0 | 1 | 2 | 3 | 4 | 14 | 6 | 12 | 11 | 10 | 9 | 8 | 7 | 13 | 5 | 15 | 16 | 21 | 18 | 23 | 20 | 17 | 22 | 19 |

