1. data

data 1

```
In[1]:= S1 = {11 → 17, 17 → 19, 19 → 13, 13 → 11, 14 → 18, 18 → 16, 16 → 12, 12 → 14, 31 → 61, 61 → 43,
           43 → 53, 53 → 31, 33 → 63, 63 → 41, 41 → 51, 51 → 33, 32 → 62, 62 → 42, 42 → 52, 52 → 32};
        S2 = {34 → 64, 64 → 46, 46 → 56, 56 → 34, 35 → 65, 65 → 45, 45 → 55,
           55 → 35, 36 → 66, 66 → 44, 44 → 54, 54 → 36};
        S3 = {21 → 27, 27 → 29, 29 → 23, 23 → 21, 24 → 28, 28 → 26, 26 → 22, 22 → 24, 37 → 67, 67 → 49,
           49 → 59, 59 → 37, 39 → 69, 69 → 47, 47 → 57, 57 → 39, 38 → 68, 68 → 48, 48 → 58, 58 → 38};
        S4 = {51 → 57, 57 → 59, 59 → 53, 53 → 51, 54 → 58, 58 → 56, 56 → 52, 52 → 54, 11 → 31, 31 → 27,
           27 → 47, 47 → 11, 17 → 37, 37 → 21, 21 → 41, 41 → 17, 14 → 34, 34 → 24, 24 → 44, 44 → 14};
        S5 = {12 → 32, 32 → 28, 28 → 48, 48 → 12, 18 → 38, 38 → 22, 22 → 42,
           42 → 18, 15 → 35, 35 → 25, 25 → 45, 45 → 15};
        S6 = {61 → 67, 67 → 69, 69 → 63, 63 → 61, 64 → 68, 68 → 66, 66 → 62, 62 → 64, 13 → 33, 33 → 29,
           29 → 49, 49 → 13, 19 → 39, 39 → 23, 23 → 43, 43 → 19, 16 → 36, 36 → 26, 26 → 46, 46 → 16};
        S7 = {31 → 33, 33 → 39, 39 → 37, 37 → 31, 32 → 36, 36 → 38, 38 → 34, 34 → 32, 17 → 61, 61 → 29,
           29 → 57, 57 → 17, 19 → 67, 67 → 27, 27 → 51, 51 → 19, 18 → 64, 64 → 28, 28 → 54, 54 → 18};
        S8 = {14 → 62, 62 → 26, 26 → 58, 58 → 14, 16 → 68, 68 → 24, 24 → 52,
           52 → 16, 15 → 65, 65 → 25, 25 → 55, 55 → 15};
        S9 = {41 → 43, 43 → 49, 49 → 47, 47 → 41, 42 → 46, 46 → 48, 48 → 44, 44 → 42, 11 → 63, 63 → 23,
           23 → 59, 59 → 11, 13 → 69, 69 → 21, 21 → 53, 53 → 13, 12 → 66, 66 → 22, 22 → 56, 56 → 12};
        StartSet = {31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48,
           49, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 63, 64, 65, 66, 67, 68, 69,
           11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29};
        Exch[list_] := Module[{len = Length[list[[1]]], per = {}},
          For[Exchi = 1, Exchi ≤ Length[list[[1]]], Exchi++, If[list[[1, Exchi]] ≠ list[[2, Exchi]],
            per = per ⋃ {list[[1, Exchi]] → list[[2, Exchi]]}]];
          per
         ]
        IS1 = Exch[{StartSet, StartSet /. S1 /. S1 /. S1}];
        IS2 = Exch[{StartSet, StartSet /. S2 /. S2 /. S2}];
        IS3 = Exch[{StartSet, StartSet /. S3 /. S3 /. S3}];
        IS4 = Exch[{StartSet, StartSet /. S4 /. S4 /. S4}];
        IS5 = Exch[{StartSet, StartSet /. S5 /. S5 /. S5}];
        IS6 = Exch[{StartSet, StartSet /. S6 /. S6 /. S6}];
        IS7 = Exch[{StartSet, StartSet /. S7 /. S7 /. S7}];
        IS8 = Exch[{StartSet, StartSet /. S8 /. S8 /. S8}];
        IS9 = Exch[{StartSet, StartSet /. S9 /. S9 /. S9}];
        prmt(*mypermutation*) =
          {S1, S2, S3, S4, S5, S6, S7, S8, S9, IS1, IS2, IS3, IS4, IS5, IS6, IS7, IS8, IS9};

        T1 = Exch[{StartSet, StartSet /. S8 /. S3 /. IS8}](*S8^-1S3 S8*);
        IT1 = Exch[{StartSet, StartSet /. T1 /. T1 /. T1}];
        T2 = Exch[{StartSet, StartSet /. S7 /. S3 /. IS7}](*S7^-1 S3 S7*);
        IT2 = Exch[{StartSet, StartSet /. T2 /. T2 /. T2}];
        T3 = Exch[{StartSet, StartSet /. S5 /. S3 /. IS5}](*S5^-1 S3 S5*);
        IT3 = Exch[{StartSet, StartSet /. T3 /. T3 /. T3}];
        T4 = Exch[{StartSet, StartSet /. S5 /. S3 /. IS5 /. S3 /. S3 /. S5 /. S3 /. IS5}]
         (*S5^-1S3 S5 S3^2 S5^-1S3 S5*);
        IT4 = Exch[{StartSet, StartSet /. T4 /. T4}];
        T5 = Exch[{StartSet, StartSet /. S6 /. S3 /. S3 /. IS6 /. S3 /. S6 /. S3 /. IS6}]
```

```
 (*S6^-1S3 S6 S3 S6^-1S3^2S6*);
IT5 = Exch[{StartSet, StartSet /. T5 /. T5 /. T5 /. T5 /. T5}];
T6 = Exch[{StartSet, StartSet /. S4 /. S5 /. S6 /. T4 /. IS4 /. IS5 /. IS6}]
 (*S4^-1S5^-1S6^-1S5^-1S3 S5 S3^2 S5^-1S3 S5 S4 S5 S6*);
IT6 = Exch[{StartSet, StartSet /. T6 /. T6}];
T7 = Exch[{StartSet,
    StartSet /. IS1 /. IS2 /. IS3 /. S4 /. S5 /. S6 /. T4 /. IS4 /. IS5 /. IS6 /. S1 /. S2 /. S3}]
   (*S1 S2 S3 S4^-1S5^-1S6^-1S5^-1S3 S5 S3^2 S5^-1S3 S5 S4 S5 S6 S1^-1S2^-1S3^-1*);
IT7 = Exch[{StartSet, StartSet /. T7 /. T7}];
T8 =
  Exch[{StartSet, StartSet /. S1 /. S1 /. S2 /. S2 /. S3 /. S3 /. S4 /. S5 /. S6 /. T4 /. IS4 /.
          IS5 /. IS6 /. S1 /. S1 /. S2 /. S2 /. S3 /. S3}]
   (*S1^2 S2^2 S3^2 S4^-1S5^-1S6^-1S5^-1S3 S5 S3^2 S5^-1S3 S5 S4 S5 S6 S1^2S2^2S3^2*);
IT8 = Exch[{StartSet, StartSet /. T8 /. T8}];
T9 = Exch[{StartSet, StartSet /. T8 /. T8 /. S3 /. T8}]
 (*T8 S3 T8^-1=S1^2 S2^2 S3^2 S4^-1S5^-1S6^-1S5^-1S3 S5 S3^2 S5^-1S3 S5 S4 S5 S6 S1^2S2^2S3^2
   S3 S1^2S2^2S3^2S4^-1S5^-1S6^-1S5^-1S3^-1S5 S3^2S5^-1S3^-1S5  S4 S5 S6 S1^2 S2^2 S3^2 *);
IT9 = Exch[{StartSet, StartSet /. T9 /. T9 /. T9}];
T10 = Exch[{StartSet, StartSet /. S6 /. S9 /. IS3 /. IS9 /. S3 /. IS6}]
 (*S6^-1S3 S9^-1S3^-1S9 S6*);
IT10 = Exch[{StartSet, StartSet /. T10 /. T10 /. T10 /. T10 /. T10}];
T11 = Exch[{StartSet,
    StartSet /. S6 /. S3 /. S3 /. IS6 /. S3 /. S6 /. IS3 /. IS6 /. S3 /. S6 /. S3 /. IS6}]
   (*S6^-1S3 S6 S3 S6^-1S3^-1S6 S3 S6^-1S3^2S6*);
IT11 = Exch[{StartSet, StartSet /. T11 /. T11}];
T12 = Exch[{StartSet, StartSet /. S6 /. S3 /. T4 /. T10 /. IS6}]
 (*S6^-1T10 T4 S3 S6=S6^-1S6^-1S3 S9^-1S3^-1S9 S6 S5^-1S3 S5 S3^2 S5^-1S3 S5 S3 S6*);
IT12 = Exch[{StartSet,
    StartSet /. T12 /. T12 /. T12 /. T12 /. T12 /. T12 /. T12 /. T12 /. T12 /. T12 /. T12}];
T13 = Exch[{StartSet, StartSet /. T12 /. T11}](*T11 T12=S6^-1S3 S6 S3 S6^-1S3^-1
   S6 S3 S6^-1S3^2S6 S6^-1S6^-1S3 S9^-1S3^-1S9 S6 S5^-1S3 S5 S3^2 S5^-1S3 S5 S3 S6*);
IT13 = Exch[{StartSet, StartSet /. T13 /. T13 /. T13 /. T13 /. T13 /. T13 /. T13 /. T13 /. T13 /.
      T13 /. T13}];
T14 = Exch[{StartSet, StartSet /. T10 /. IS3}](*S3^-1T10=S3^-1S6^-1S3 S9^-1S3^-1S9 S6*);
IT14 = Exch[{StartSet,
    StartSet /. T14 /. T14 /. T14 /. T14 /. T14 /. T14 /. T14 /. T14 /. T14 /. T14 /. T14}];
T15 = Exch[{StartSet, StartSet /. S3 /. S3 /. T14 /. T5}]
 (*T5 T14 S3^2=S6^-1S3 S6 S3 S6^-1S3^2S6 S3^-1S6^-1S3 S9^-1S3^-1S9 S6 S3^2*);
IT15 = Exch[{StartSet,
    StartSet /. T15 /. T15 /. T15 /. T15 /. T15 /. T15 /. T15 /. T15 /. T15 /. T15 /. T15}];
T16 = Exch[{StartSet, StartSet /. T5 /. T5 /. T12}](*T12 T5^2=S6^-1S6^-1S3 S9^-1S3^-1S9
    S6 S5^-1S3 S5 S3^2 S5^-1S3 S5 S3 S6 S6^-1S3 S6 S3 S6^-1S3^2S6 S6^-1S3 S6 S3 S6^-1S3^2S6=
   S6^-1S6^-1S3 S9^-1S3^-1S9 S6 S5^-1S3 S5 S3^2 S5^-1S3 S5 S3 S3 S6 S3 S6^-1S3^-1 S6 S3 S6^-1S3^2S6*);
IT16 = Exch[{StartSet, StartSet /. T16 /. T16 /. T16 /. T16 /. T16 /. T16 /. T16 /. T16 /. T16 /.
      T16 /. T16}];
(*IT13=Exch[{StartSet,StartSet/.IT11/.IT12}];
IT14=Exch[{StartSet,StartSet/.S3/.IT10}];
IT15=Exch[{StartSet,StartSet/.IT5/.IT14/.S3/.S3}];
IT16=Exch[{StartSet,StartSet/.IT12/.IT5/.IT5}];*)
T17 = Exch[{StartSet, StartSet /. T16 /. IT13}](*T13^-1T16=T12^-1T11^-1T12 T5^2*);
IT17 = Exch[{StartSet, StartSet /. T17 /. T17 /. T17 /. T17 /. T17}];
T18 = Exch[{StartSet, StartSet /. IT15 /. T14}](*T14 T15^-1=S3^-1T10 S3^2 T14^-1T5^-1*);
IT18 = Exch[{StartSet, StartSet /. T18 /. T18}];
T19 = Exch[{StartSet, StartSet /. T14 /. T14 /. IT13}]
 (*T13^-1T14^2=T12^-1T11^-1S3^-1T10 S3^-1T10*);
IT19 = Exch[{StartSet, StartSet /. T19 /. T19 /. T19}];
T20 = Exch[{StartSet, StartSet /. T19 /. T19}]
```

```
 (*T19^2=T12^-1T11^-1S3^-1T10 S3^-1T10 T12^-1T11^-1S3^-1T10 S3^-1T10*);
IT20 = Exch[{StartSet, StartSet /. T20}];
T21 = Exch[{StartSet, StartSet /. T17 /. T17 /. T17}]
 (*T17^3=T12^-1T11^-1T12 T5^2T12^-1T11^-1T12 T5^2T12^-1T11^-1T12 T5^2*);
IT21 = Exch[{StartSet, StartSet /. T21}];
T22 = Exch[
   {StartSet, StartSet /. S1 /. S1 /. S2 /. S2 /. S3 /. S3 /. T4 /. S1 /. S1 /. S2 /. S2 /. S3 /.
     S3}](*S1^2S2^2S3^2T4 S1^2S2^2S3^2*);
IT22 = Exch[{StartSet, StartSet /. T22 /. T22}];
T23 = Exch[{StartSet, StartSet /. T4 /. T20 /. T4 /. T22}]
 (*S1^2S2^2S3^2T4 S1^2S2^2S3^2 T4 T20 T4*);
IT23 = Exch[{StartSet, StartSet /. T23}];
T24 = Exch[{StartSet, StartSet /. IS5 /. T20 /. S5 /. S5 /. T3 /. S3 /. T3 /. T3 /. T21}]
 (*T21 T3^2S3 T3 S5^2T20 S5^-1*);
IT24 = Exch[{StartSet, StartSet /. T24 /. T24 /. T24}];
T25 = Exch[{StartSet, StartSet /. S5 /. S5 /. S2 /. S5 /. S5 /. IS2}](*S2^-1S5^2S2 S5^2*);
IT25 = Exch[{StartSet, StartSet /. T25}];
T26 = Exch[{StartSet, StartSet /. S2 /. S2 /. S5 /. S2 /. S2 /. IS5}](*S5^-1S2^2S5 S2^2*);
IT26 = Exch[{StartSet, StartSet /. T26}];
T27 = Exch[{StartSet, StartSet /. IS1 /. IS2 /. S1 /. S5 /. S2 /. IS5}]
 (*S5^-1S2 S5 S1 S2^-1S1^-1*);
IT27 = Exch[{StartSet, StartSet /. T27 /. T27}];
```

data 2

```
In[76]:= llstrtSt = {11, 12, 13, 14, 16, 17, 18, 19, 34, 36, 44, 46, 21, 23, 27, 22, 24, 26, 15, 35};
TrnsfrmStfrGrph = {S1⋃S2⋃S3⋃S4⋃S5⋃S6⋃S7⋃S8⋃S9, S2⋃S3⋃S5⋃S6⋃S7⋃S8,
   S2⋃S3⋃S6⋃S7⋃S8, S2⋃S3⋃S7⋃S8, S2⋃S3⋃S7⋃T1, S2⋃S3⋃S7, S2⋃S3⋃T2⋃T3,
   S2⋃S3⋃T2⋃T4, S2⋃S3⋃T4⋃T5⋃T6, S3⋃T4⋃T5⋃T7⋃T8, S3⋃T4⋃T5⋃T8,
   S3⋃T4⋃T5⋃T9, S3⋃T4⋃T5⋃T10⋃T11⋃T12, T4⋃T13⋃T14⋃T15⋃T16,
   T4⋃T17⋃T18⋃T19, T4⋃T20⋃T21, T22⋃T23, T23⋃T24, T25⋃T26⋃T27, T25};
TrnsfrmStfrPrmttn = {{S1, S2, S3, S4, S5, S6, S7, S8, S9}, {S2, S3, S5, S6, S7, S8},
   {S2, S3, S6, S7, S8}, {S2, S3, S7, S8}, {S2, S3, S7, T1}, {S2, S3, S7}, {S2, S3, T2, T3},
   {S2, S3, T2, T4}, {S2, S3, T4, T5, T6}, {S3, T4, T5, T7, T8}, {S3, T4, T5, T8},
   {S3, T4, T5, T9}, {S3, T4, T5, T10, T11, T12}, {T4, T13, T14, T15, T16},
   {T4, T17, T18, T19}, {T4, T20, T21}, {T22, T23}, {T23, T24}, {T25, T26, T27}, {T25}};
TrnsfrmStfrshrtPrnt = {{"S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8", "S9"},
   {"S2", "S3", "S5", "S6", "S7", "S8"}, {"S2", "S3", "S6", "S7", "S8"},
   {"S2", "S3", "S7", "S8"}, {"S2", "S3", "S7", "T1"}, {"S2", "S3", "S7"},
   {"S2", "S3", "T2", "T3"}, {"S2", "S3", "T2", "T4"}, {"S2", "S3", "T4", "T5", "T6"},
   {"S3", "T4", "T5", "T7", "T8"}, {"S3", "T4", "T5", "T8"},
   {"S3", "T4", "T5", "T9"}, {"S3", "T4", "T5", "T10", "T11", "T12"},
   {"T4", "T13", "T14", "T15", "T16"}, {"T4", "T17", "T18", "T19"}, {"T4", "T20", "T21"},
   {"T22", "T23"}, {"T23", "T24"}, {"T25", "T26", "T27"}, {"T25"}};
TrnsfrmStfrlngPrnt = {{"S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8", "S9"},
   {"S2", "S3", "S5", "S6", "S7", "S8"}, {"S2", "S3", "S6", "S7", "S8"},
   {"S2", "S3", "S7", "S8"}, {"S2", "S3", "S7", "S8 S3 S8^-1"},
   {"S2", "S3", "S7"}, {"S2", "S3", "S7 S3 S7^-1", "S5 S3 S5^-1"},
   {"S2", "S3", "S7 S3 S7^-1", "S5 S3 S5^-1S3^2S5 S3 S5^-1"},
   {"S2", "S3", "S5 S3 S5^-1S3^2S5 S3 S5^-1", "S6 S3^2S6^-1S3 S6 S3 S6^-1",
    "S4 S5 S6 S5 S3 S5^-1S3^2S5 S3 S5^-1S6^-1S5^-1S4^-1"},
   {"S3", "S5 S3 S5^-1S3^2S5 S3 S5^-1", "S6 S3^2S6^-1S3 S6 S3 S6^-1",
    "S1^-1S2^-1S3^-1S4 S5 S6 S5 S3 S5^-1S3^2S5 S3 S5^-1S6^-1S5^-1S4^-1S1 S2 S3",
    "S1^2S2^2S3^2S4 S5 S6 S5 S3 S5^-1S3^2S5 S3 S5^-1S6^-1S5^-1S4^-1S1^2S2^2S3^2"},
    {"S3", "S5 S3 S5^-1S3^2S5 S3 S5^-1", "S6 S3^2S6^-1S3 S6 S3 S6^-1",
     "S1^2S2^2S3^2S4 S5 S6 S5 S3 S5^-1S3^2S5 S3 S5^-1S6^-1S5^-1S4^-1S1^2S2^2S3^2"},
    {"S3", "S5 S3 S5^-1S3^2S5 S3 S5^-1", "S6 S3^2S6^-1S3 S6 S3 S6^-1",
     "S1^2 S2^2 S3^2S4 S5 S6 S5 S3^-1S5^-1S3^2S5 S3^-1S5^-1S4^-1S5^-1S6^-1S1^2S2^2S3^2S3
       S1^2S2^2S3^2 S4 S5 S6 S5 S3 S5^-1S3^2 S5 S3 S5^-1 S4^-1S5^-1S6^-1S1^2 S2^2 S3^2"},
    {"S3", "S5 S3 S5^-1S3^2S5 S3 S5^-1", "S6 S3^2S6^-1S3 S6 S3 S6^-1", "S6 S9 S3^-1S9^-1S3 S6^-1",
     "S6 S3^2S6^-1S3 S6 S3^-1S6^-1S3 S6 S3 S6^-1",
     "S6 S3 S5 S3 S5^-1S3^2 S5 S3 S5^-1S6 S9 S3^-1S9^-1S3 S6^-1S6^-1"},
    {"S5 S3 S5^-1S3^2S5 S3 S5^-1", "T12 T11", "T10 S3^-1", "S3^2T14 T5", "T5^2T12"},
    {"S5 S3 S5^-1S3^2S5 S3 S5^-1", "T5^2T12 T11^-1T12^-1",
     "T5^-1T14^-1S3^2T10 S3^-1", "T10 S3^-1T10 S3^-1T11^-1T12^-1"},
    {"S5 S3 S5^-1S3^2S5 S3 S5^-1", "T10 S3^-1T10 S3^-1T11^-1T12^-1T10 S3^-1T10 S3^-1T11^-1T12^-1",
     "T5^2T12 T11^-1T12^-1T5^2T12 T11^-1T12^-1T5^2T12 T11^-1T12^-1"}, {"S1^2S2^2S3^2T4 S1^2S2^2S3^2",
     "T4 T10 S3^-1T10 S3^-1T11^-1T12^-1T10 S3^-1T10 S3^-1T11^-1T12^-1 T4 S1^2S2^2S3^2T4 S1^2S2^2S3^2"},
    {"T4 T10 S3^-1T10 S3^-1T11^-1T12^-1T10 S3^-1T10 S3^-1T11^-1T12^-1 T4 S1^2S2^2S3^2T4 S1^2S2^2S3^2",
     "T5^2T12 T11^-1T12^-1T5^2T12 T11^-1T12^-1T5^2T12 T11^-1T12^-1 T3^2S3 T3
       S5^2T10 S3^-1T10 S3^-1T11^-1T12^-1T10 S3^-1T10 S3^-1T11^-1T12^-1 S5^-1"},
    {"S2^-1S5^2S2 S5^2", "S5^-1S2^2S5 S2^2", "S1^-1S2^-1S1 S5 S2 S5^-1"}, {"S2^-1S5^2S2 S5^2"}};
TrnsfrmStfrlst = {{{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}},
    {{2}, {3}, {5}, {6}, {7}, {8}}, {{2}, {3}, {6}, {7}, {8}}, {{2}, {3}, {7}, {8}},
    {{2}, {3}, {7}, {8, 3, 17}}, {{2}, {3}, {7}}, {{2}, {3}, {7, 3, 16}, {5, 3, 14}},
    {{2}, {3}, {7, 3, 16}, {5, 3, 14, 3, 3, 5, 3, 14}}, {{2}, {3}, {5, 3, 14, 3, 3, 5, 3, 14},
     {6, 3, 3, 15, 3, 6, 3, 15}, {4, 5, 6, 5, 3, 14, 3, 3, 5, 3, 14, 15, 14, 13}},
    {{3}, {5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 3, 15, 3, 6, 3, 15},
     {10, 11, 12, 4, 5, 6, 5, 3, 14, 3, 3, 5, 3, 14, 15, 14, 13, 1, 2, 3},
     {1, 1, 2, 2, 3, 3, 4, 5, 6, 5, 3, 14, 3, 3, 5, 3, 14, 15, 14, 13, 1, 1, 2, 2, 3, 3}},
    {{3}, {5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 3, 15, 3, 6, 3, 15},
     {1, 1, 2, 2, 3, 3, 4, 5, 6, 5, 3, 14, 3, 3, 5, 3, 14, 15, 14, 13, 1, 1, 2, 2, 3, 3}},
    {{3}, {5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 3, 15, 3, 6, 3, 15},
     {1, 1, 2, 2, 3, 3, 4, 5, 6, 5, 12, 14, 3, 3, 5, 12, 14, 13, 14, 15, 1, 1, 2, 2, 3, 3, 3,
      1, 1, 2, 2, 3, 3, 4, 5, 6, 5, 3, 14, 3, 3, 5, 3, 14, 13, 14, 15, 1, 1, 2, 2, 3, 3}},
    {{3}, {5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 3, 15, 3, 6, 3, 15}, {6, 9, 12, 18, 3, 15}, {6, 3, 3,
      15, 3, 6, 12, 15, 3, 6, 3, 15}, {6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15}},
    {{5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3,
      15, 15, 6, 3, 3, 15, 3, 6, 12, 15, 3, 6, 3, 15}, {6, 9, 12, 18, 3, 15, 12},
     {3, 3, 6, 9, 12, 18, 3, 15, 12, 6, 3, 3, 15, 3, 6, 3, 15}, {6, 3, 3, 15, 3, 6, 3, 15,
      6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15}},
    {{5, 3, 14, 3, 3, 5, 3, 14}, {6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 3, 15, 3, 6, 3, 15,
      6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15, 6, 12, 15, 12, 6, 3, 15,
      12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15},
     {6, 12, 15, 12, 6, 12, 12, 15, 3, 6, 12, 9, 3, 18, 15 , 3, 3, 6, 9, 12, 18, 3, 15, 12},
     {6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3, 15, 12,
      6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15}},
    {{5, 3, 14, 3, 3, 5, 3, 14}, {6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6,
      12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5,
      12, 14, 12, 15, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3,
      15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15},
     {6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9,
      12, 18, 3, 15, 15, 6, 12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12,
      14, 12, 12, 5, 12, 14, 12, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 5,
      3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15, 6, 12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15,
      6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3,
      3, 15, 3, 6, 3, 15, 6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15, 6, 12, 15, 12,
      6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15}},
    {{1, 1, 2, 2, 3, 3, 5, 3, 14, 3, 3, 5, 3, 14, 1, 1, 2, 2, 3, 3},
 {5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15,
  12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14,
  12, 15, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3, 15,
  12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 5,
  3, 14, 3, 3, 5, 3, 14, 1, 1, 2, 2, 3, 3, 5, 3, 14, 3, 3, 5, 3, 14, 1, 1, 2, 2, 3, 3}},
{{5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12,
  15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12,
  14, 12, 15, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3, 15,
  12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 5,
  3, 14, 3, 3, 5, 3, 14, 1, 1, 2, 2, 3, 3, 5, 3, 14, 3, 3, 5, 3, 14, 1, 1, 2, 2, 3, 3},
 {14, 6, 9, 12, 18, 3, 15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3, 15, 12, 6,
  12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 6, 9, 12, 18, 3,
  15, 12, 6, 9, 12, 18, 3, 15, 12, 6, 12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3,
  18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 5, 5, 5, 3, 14, 3, 5, 3, 14, 5, 3, 14, 6, 3, 3,
  15, 3, 6, 3, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3,
  15, 15, 6, 12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12,
  12, 5, 12, 14, 12, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 5, 3, 14,
  3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15, 6, 12, 15, 12, 6, 3, 15, 12, 6, 12, 12, 15, 6, 6,
  12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15, 6, 3, 3, 15, 3, 6, 3, 15, 6, 3, 3,
  15, 3, 6, 3, 15, 6, 3, 5, 3, 14, 3, 3, 5, 3, 14, 6, 9, 12, 18, 3, 15, 15, 6, 12, 15, 12, 6,
  3, 15, 12, 6, 12, 12, 15, 6, 6, 12, 9, 3, 18, 15, 5, 12, 14, 12, 12, 5, 12, 14, 12, 15}},
{{11, 5, 5, 2, 5, 5}, {14, 2, 2, 5, 2, 2}, {10, 11, 1, 5, 2, 14}}, {{11, 5, 5, 2, 5, 5}}};
```

data 3

```
In[82]:= square = {{{Glow[Blue], Black, Cuboid[{0, 2, 3}, {1, 3, 3.001}]},
             {Glow[Blue], Black, Cuboid[{1, 2, 3}, {2, 3, 3.001}]},
             {Glow[Blue], Black, Cuboid[{2, 2, 3}, {3, 3, 3.001}]},
             {Glow[Blue], Black, Cuboid[{0, 1, 3}, {1, 2, 3.001}]},
             {Glow[Blue], Black, Cuboid[{1, 1, 3}, {2, 2, 3.001}]},
             {Glow[Blue], Black, Cuboid[{2, 1, 3}, {3, 2, 3.001}]},
             {Glow[Blue], Black, Cuboid[{0, 0, 3}, {1, 1, 3.001}]},
             {Glow[Blue], Black, Cuboid[{1, 0, 3}, {2, 1, 3.001}]},
             {Glow[Blue], Black, Cuboid[{2, 0, 3}, {3, 1, 3.001}]}},
            {{Glow[Yellow], Black, Cuboid[{0, 2, 0}, {1, 3, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{1, 2, 0}, {2, 3, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{2, 2, 0}, {3, 3, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{0, 1, 0}, {1, 2, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{1, 1, 0}, {2, 2, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{2, 1, 0}, {3, 2, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{0, 0, 0}, {1, 1, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{1, 0, 0}, {2, 1, 0.001}]},
             {Glow[Yellow], Black, Cuboid[{2, 0, 0}, {3, 1, 0.001}]}},
            {{Glow[Green], Black, Cuboid[{0, 0, 2}, {1, 0.001, 3}]},
             {Glow[Green], Black, Cuboid[{1, 0, 2}, {2, 0.001, 3}]},
             {Glow[Green], Black, Cuboid[{2, 0, 2}, {3, 0.001, 3}]},
             {Glow[Green], Black, Cuboid[{0, 0, 1}, {1, 0.001, 2}]},
             {Glow[Green], Black, Cuboid[{1, 0, 1}, {2, 0.001, 2}]},
             {Glow[Green], Black, Cuboid[{2, 0, 1}, {3, 0.001, 2}]},
             {Glow[Green], Black, Cuboid[{0, 0, 0}, {1, 0.001, 1}]},
             {Glow[Green], Black, Cuboid[{1, 0, 0}, {2, 0.001, 1}]},
             {Glow[Green], Black, Cuboid[{2, 0, 0}, {3, 0.001, 1}]}},
            {{Glow[Purple], Black, Cuboid[{0, 3, 2}, {1, 3.001, 3}]},
             {Glow[Purple], Black, Cuboid[{1, 3, 2}, {2, 3.001, 3}]},
             {Glow[Purple], Black, Cuboid[{2, 3, 2}, {3, 3.001, 3}]},
             {Glow[Purple], Black, Cuboid[{0, 3, 1}, {1, 3.001, 2}]},
             {Glow[Purple], Black, Cuboid[{1, 3, 1}, {2, 3.001, 2}]},
             {Glow[Purple], Black, Cuboid[{2, 3, 1}, {3, 3.001, 2}]},
             {Glow[Purple], Black, Cuboid[{0, 3, 0}, {1, 3.001, 1}]},
             {Glow[Purple], Black, Cuboid[{1, 3, 0}, {2, 3.001, 1}]},
             {Glow[Purple], Black, Cuboid[{2, 3, 0}, {3, 3.001, 1}]}},
            {{Glow[Red], Black, Cuboid[{0, 0, 2}, {0.001, 1, 3}]},
             {Glow[Red], Black, Cuboid[{0, 1, 2}, {0.001, 2, 3}]},
             {Glow[Red], Black, Cuboid[{0, 2, 2}, {0.001, 3, 3}]}, {Glow[Red], Black,
              Cuboid[{0, 0, 1}, {0.001, 1, 2}]}, {Glow[Red], Black, Cuboid[{0, 1, 1}, {0.001, 2, 2}]},
             {Glow[Red], Black, Cuboid[{0, 2, 1}, {0.001, 3, 2}]}, {Glow[Red], Black,
              Cuboid[{0, 0, 0}, {0.001, 1, 1}]}, {Glow[Red], Black, Cuboid[{0, 1, 0}, {0.001, 2, 1}]},
             {Glow[Red], Black, Cuboid[{0, 2, 0}, {0.001, 3, 1}]}},
            {{Glow[Orange], Black, Cuboid[{3, 0, 2}, {3.001, 1, 3}]},
             {Glow[Orange], Black, Cuboid[{3, 1, 2}, {3.001, 2, 3}]},
             {Glow[Orange], Black, Cuboid[{3, 2, 2}, {3.001, 3, 3}]},
             {Glow[Orange], Black, Cuboid[{3, 0, 1}, {3.001, 1, 2}]},
             {Glow[Orange], Black, Cuboid[{3, 1, 1}, {3.001, 2, 2}]},
             {Glow[Orange], Black, Cuboid[{3, 2, 1}, {3.001, 3, 2}]},
             {Glow[Orange], Black, Cuboid[{3, 0, 0}, {3.001, 1, 1}]},
             {Glow[Orange], Black, Cuboid[{3, 1, 0}, {3.001, 2, 1}]},
             {Glow[Orange], Black, Cuboid[{3, 2, 0}, {3.001, 3, 1}]}}};
```

2. function

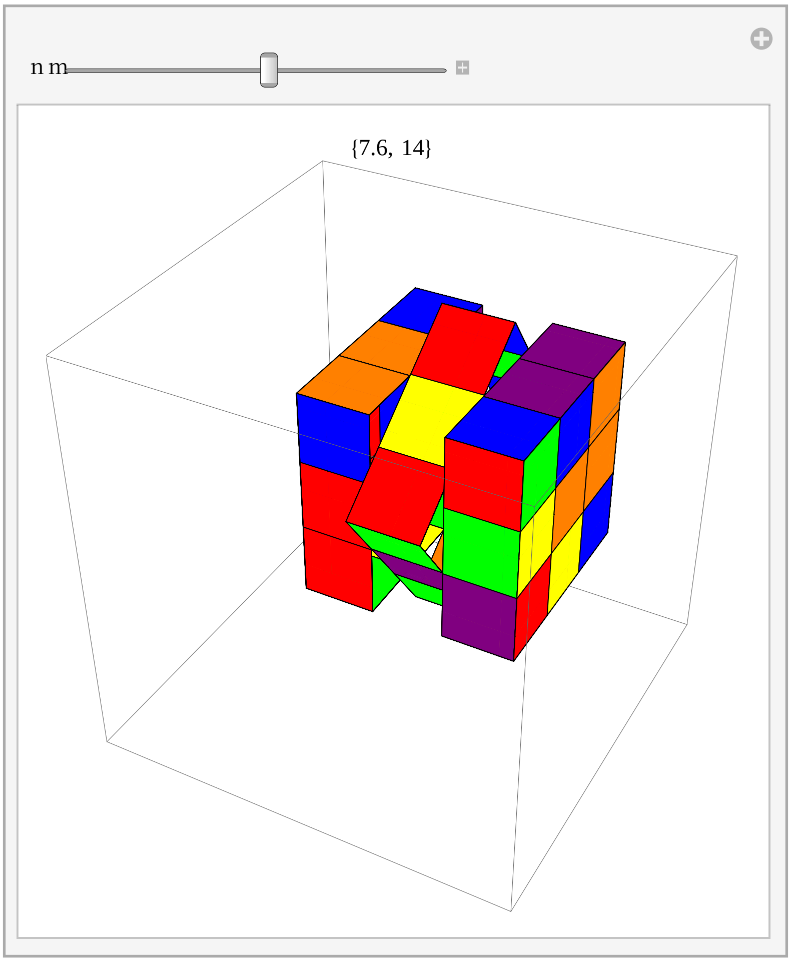
```
In[83]:= (*提取变换前后的集合*)
        ExtractSet[StartSet_, Tans_] :=
          Module[{StrtSt = StartSet, ndSt = {}, myi = 1}, ndSt = StrtSt /. Tans;
           While[myi ≤ Length[StrtSt], If[StrtSt[[myi]] == ndSt[[myi]],
             StrtSt = Delete[StrtSt, myi]; ndSt = Delete[ndSt, myi]; myi--]; myi++];
           {StrtSt, ndSt}
          ];
        Vsl[StartP_] := Module[{nP = {}, pthfrshrtPrnt = {},
            pthfrlngPrnt = {}, pthfrlst = {}, g = {}, pth = {}, lst = {}},
           nP = StartP;
           For[i = 1, i ≤ Length[llstrtSt], i++, g = Graph[TrnsfrmStfrGrph[[i]],
              VertexLabels → "Name", GraphStyle → "SmallNetwork", VertexSize → Small,
              DirectedEdges → True, GraphLayout → "CircularEmbedding", ImageSize → 450];
            pth = FindShortestPath[g, llstrtSt[[i]] /. nP, llstrtSt[[i]]]; (*用于消元的路径*)
            lst = If[pth ≠ {},
              Flatten[Table[Select[Table[ifrntgr, {ifrntgr, Length[TrnsfrmStfrPrmttn[[i]]]}],
                 (pth[[ifrpth]] /. TrnsfrmStfrPrmttn[[i, #]]) == pth[[ifrpth + 1]] &,
                 1], {ifrpth, Length[pth] - 1}]], {}];
            (*lst: nP/.TrnsfrmStfrPrnt[[i,lst1]]/.TrnsfrmStfrPrnt[[i,lst2]]...*)
            For[ifrnP = 1, ifrnP ≤ Length[lst], ifrnP++,
             nP = Exch[{StartSet, StartSet /. nP /. TrnsfrmStfrPrmttn[[i, lst[[ifrnP]]]]}];
             AppendTo[pthfrshrtPrnt, TrnsfrmStfrshrtPrnt[[i, lst[[ifrnP]]]]];
             AppendTo[pthfrlngPrnt, TrnsfrmStfrlngPrnt[[i, lst[[ifrnP]]]]];
             AppendTo[pthfrlst, TrnsfrmStfrlst[[i, lst[[ifrnP]]]]]];
           ];
           Flatten[pthfrlst]
          ];
        pdtVsl[StartP_] := Module[{lst = Vsl[StartP], ifrlst = 1, nwlst = {}, bl = True},
           While[bl,
            While[ifrlst ≤ Length[lst],
             nwlst = Vsl[PermulateMultiply[StartSet /. StartP, lst[[1 ;; ifrlst]]]];
             If[Length[nwlst] < Length[lst] - ifrlst,
              lst = Flatten[{lst[[1 ;; ifrlst]], nwlst}], ifrlst++];
            ];
            If[lst === Dlt[lst], bl = False];
            If[bl, While[lst =!= Dlt[lst], lst = Dlt[lst]]];
           ];
           lst
          ];
        TS[ifrS_][sqr_, nm_] := Module[{sq = sqr, lst = {}},
           If[ifrS == 1 || ifrS == 10,
            lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃ {{1, 5}}];
           If[ifrS == 3 || ifrS == 12, lst =
             IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃ {{2, 5}}];
           If[ifrS == 4 || ifrS == 13, lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃
              {{5, 5}}];
           If[ifrS == 6 || ifrS == 15, lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃
              {{6, 5}}];
           If[ifrS == 7 || ifrS == 16, lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃
              {{3, 5}}];
           If[ifrS == 9 || ifrS == 18, lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]] ⋃
              {{4, 5}}];
           If[ifrS == 2 || ifrS == 5 || ifrS == 8 || ifrS == 11 || ifrS == 14 || ifrS == 17,
            lst = IntegerDigits[ExtractSet[StartSet, prmt[[ifrS]]]][[1]]];
           Do[sq[[lst[[ifrD, 1]], lst[[ifrD, 2]], 3]] =
      Rotate[sqr[[lst[[ifrD, 1]], lst[[ifrD, 2]], 3]], Sign[9.5 - ifrS] 90 nm Degree,
        {Boole[4 ≤ ifrS ≤ 6 || 13 ≤ ifrS ≤ 15], Boole[7 ≤ ifrS ≤ 9 || 16 ≤ ifrS ≤ 18],
         Boole[1 ≤ ifrS ≤ 3 || 10 ≤ ifrS ≤ 12]}, {3/2, 3/2, 3/2}]], {ifrD, Length[lst]}];
    sq];
PermulateMultiply[Srtst_, List_] := Module[{Cntr = Srtst, ndSt = {}},
    For[i = 1, i ≤ Length[List], i++, Cntr = Cntr /. prmt[[List[[i]]]]];
    ndSt = Exch[{StartSet, Cntr}]];(*Srtst:StartSet, List: AntTrace[...]*)
TImdtl[sqr_, prmttn_] := Module[{sq = sqr, keepp = {}, xtrctSt = {}, i = 1, j = 1},
    xtrctSt = IntegerDigits[ExtractSet[StartSet, prmttn]];
    For[i = 1, i ≤ Length[xtrctSt[[1]]], i++,
     AppendTo[keepp, sq[[xtrctSt[[1, i, 1]], xtrctSt[[1, i, 2]], 1]]]];
    For[i = 1, i ≤ Length[xtrctSt[[1]]], i++,
     sq[[xtrctSt[[2, i, 1]], xtrctSt[[2, i, 2]], 1]] = keepp[[i]]];
    sq];
TCntnS[sqr_, lst_, nm_] :=
   Module[{sq = sqr, ifrTImdtl = Floor[nm], nmfrTS = nm - Floor[nm], prmttn = {}},
    prmttn = PermulateMultiply[StartSet, lst[[1 ;; ifrTImdtl]]];
    sq = TImdtl[sq, prmttn];
    sq = TS[lst[[Min[ifrTImdtl + 1, Length[lst]]]]][sq, nmfrTS];
    sq
   ];
Dltlst[lst_] := Module[
    {ls = lst, ilst = Select[DeleteDuplicates[lst, Mod[#1 - #2, 9] == 0 &], # ≤ 9 &], ifr = 1},
    While[ifr ≤ Length[ilst],
     If[Position[ls, ilst[[ifr]]] ≠ {} && Position[ls, ilst[[ifr]] + 9] ≠ {},
       ls = Delete[ls,
         {First[Position[ls, ilst[[ifr]]]], First[Position[ls, ilst[[ifr]] + 9]]}], ifr++
      ];
    ];
    ls
   ];
Dlt[lst_] := Module[{ifrDlt = 1, ls =
      Flatten[Dltlst /@ Sort /@ Split[lst, IntegerPart[If[#1 > 9, (#1 - 10) / 3, (#1 - 1) / 3]] ==
            IntegerPart[If[#2 > 9, (#2 - 10) / 3, (#2 - 1) / 3]] &]]},
    While[ifrDlt < Length[ls],
     If[Abs[ls[[ifrDlt]] - ls[[ifrDlt + 1]]] == 9,
      ls = Delete[ls, {{ifrDlt}, {ifrDlt + 1}}]; ifrDlt--];
     ifrDlt++];
    ifrDlt = 1;
    While[ifrDlt < Length[ls] - 1,
     If[ls[[ifrDlt]] == ls[[ifrDlt + 1]] == ls[[ifrDlt + 2]],
      ls = Delete[ls, {{ifrDlt + 1}, {ifrDlt + 2}}];
      If[ls[[ifrDlt]] ≤ 9, ls[[ifrDlt]] += 9, ls[[ifrDlt]] -= 9]; ifrDlt--];
     ifrDlt++];
    ls
   ];
```

3. Main Process

First Process for each step

In[92]:=
```
StartP = Exch[{StartSet, StartSet /. S8 /. S5 /. S1 /. S2}];
list = Vsl[StartP];
While[list =!= Dlt[list], list = Dlt[list]];
Manipulate[Pause[0.1]; Graphics3D[
  TCntnS[TIMdtl[square, StartP], list, nm], PlotRange → {{-2, 4}, {-2, 4}, {-2, 4}},
  PlotLabel → {nm, Length[list]}], {nm, 0, Length[list], 0.1}]
```

Out[95]=

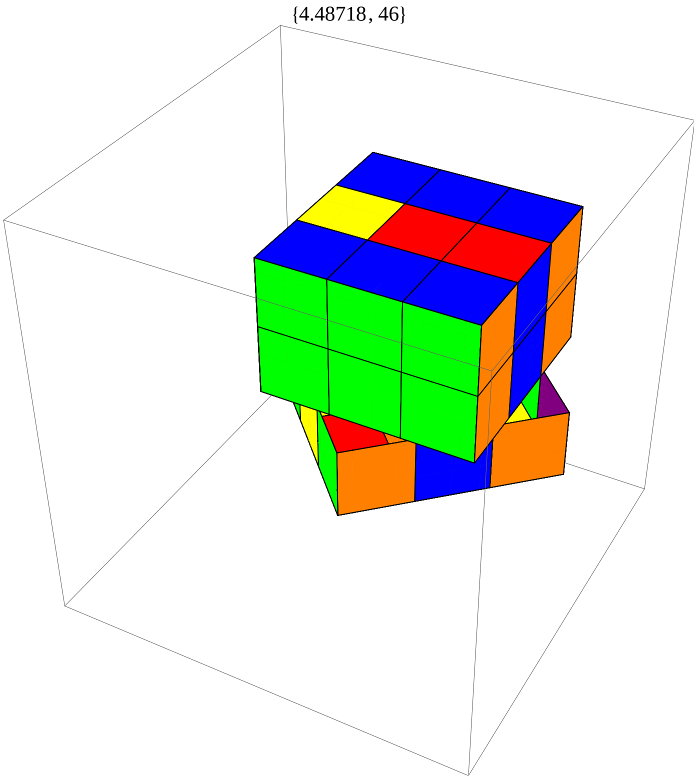


Second Process which is Automatic

```
In[103]:= StartP = Exch[{StartSet, StartSet /. S8 /. S3 /. IS8 /. S2 /. S5 /. S7 /. S8 /. S1 /. S9 /. S6}];
          nm = 0;
          list = Vsl[StartP];
          While[list =!= Dlt[list], list = Dlt[list]];
          ProgressIndicator[Dynamic[nm / Length[list]]]
          Dynamic[Graphics3D[TCntnS[TImdtl[square, StartP], list, nm],
            PlotRange → {{-2, 4}, {-2, 4}, {-2, 4}}, PlotLabel → {nm, Length[list]}]]
          Mouseover[stop,
           Table[With[{i = i}, nm = i; Pause[0.01]], {i, 0, Length[list], 0.1}]]
```

Out[107]=

Out[108]=



```
Out[109]= $Aborted
```

Third Process which is too much Manipulate

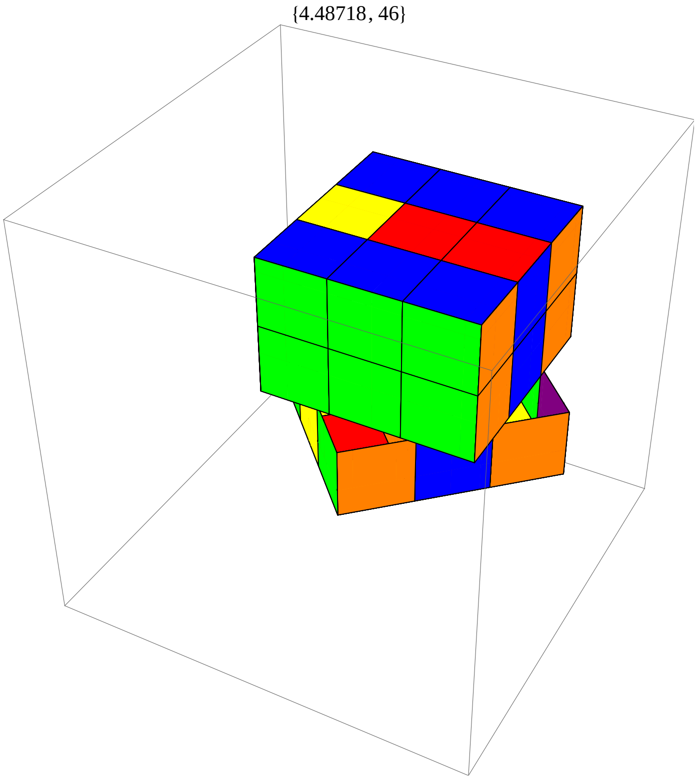
```
In[117]:= Dynamic[t = MousePosition["WindowScaled"]]
          Dynamic[t]

Out[117]= {0.506429, 0.460905}

Out[118]= {0.506429, 0.460905}
```

In[119]:=

```
StartP = Exch[{StartSet, StartSet /. S8 /. S3 /. IS8 /. S3}];
nm = 0;
list = Vsl[StartP];
While[list =!= Dlt[list], list = Dlt[list]];
ProgressIndicator[Dynamic[nm / Length[list]]]
Dynamic[Graphics3D[TCntnS[TImdtl[square, StartP], list, nm],
  PlotRange → {{-2, 4}, {-2, 4}, {-2, 4}}, PlotLabel → {nm, Length[list]}]]
Mouseover[Button[Dynamic[nm = Min[Max[nm + (2 t[[1]] - 1), 0], Length[list]]]],
 Button["Stop"]]
```

Out[123]=

Out[124]=



Out[125]=

```
4 . 5 0
```

Forth Process which is updated from last one

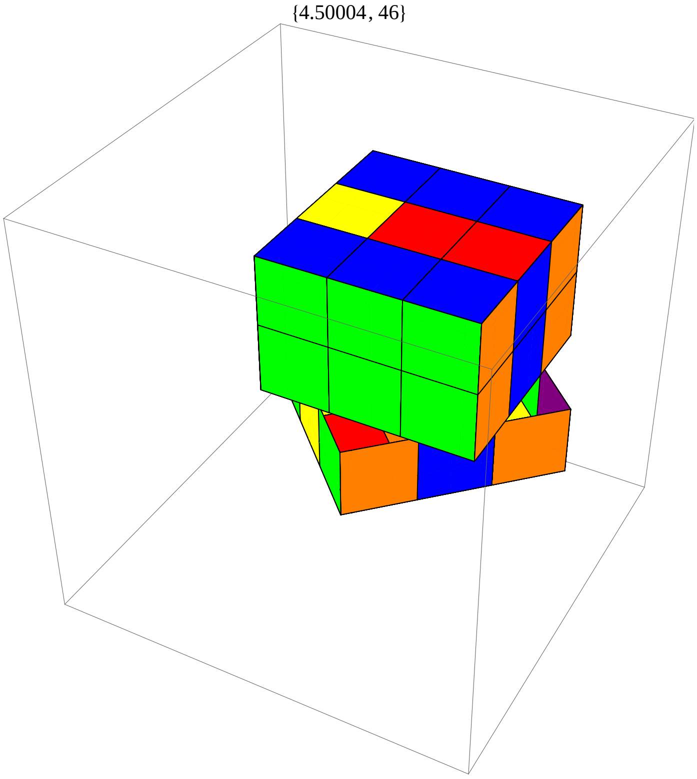
In[126]:=

```
StartP = Exch[{StartSet, StartSet /. S8 /. S3 /. IS8 /. S3 /. S6 /. S1}];
nm = 0;
list = pdtVsl[StartP];
Print[Length[Vsl[StartP]], "->", Length[list]];
ProgressIndicator[Dynamic[nm / Length[list]]]
Dynamic[Graphics3D[TCntnS[TImdtl[square, StartP], list, nm],
  PlotRange → {{-2, 4}, {-2, 4}, {-2, 4}}, PlotLabel → {nm, Length[list]}]]
Mouseover[Button[Dynamic[nm = Min[Max[nm + (2 t[[1]] - 1), 0], Length[list]]]],
 Button["Stop"]]
```

```
66->46
```

Out[130]=

Out[131]=



Out[132]=

```
4 . 5
```

```
Dynamic[t = MousePosition["WindowScaled"]]
Dynamic[t]
```

```
{0.839763, 0.394663}
```

```
{0.839763, 0.394663}
```