


```
> ( 9,11,16,14)(10,13,15,12)( 1,17,41,40)( 4,20,44,37)( 6,22,46,35),
> (17,19,24,22)(18,21,23,20)( 6,25,43,16)( 7,28,42,13)( 8,30,41,11),
> (25,27,32,30)(26,29,31,28)( 3,38,43,19)( 5,36,45,21)( 8,33,48,24),
> (33,35,40,38)(34,37,39,36)( 3, 9,46,32)( 2,12,47,29)( 1,14,48,27),
> (41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40) );
<permutation group with 6 generators>
```

First we want to know the size of this group.

```
gap> Size( cube );
43252003274489856000
```

Since this is a little bit unhandy, let us factorize this number.

```
gap> Collected( Factors( last ) );
[ [ 2, 27 ], [ 3, 14 ], [ 5, 3 ], [ 7, 2 ], [ 11, 1 ] ]
```

(The result tells us that the size is $2^{27} 3^{14} 5^3 7^2 11$.)

Next let us investigate the operation of the group on the 48 points (we reduce the line length to get a more appropriate output format).

```
gap> SizeScreen( [71, ] );;
gap> orbits := Orbits( cube, [1..48] );
[ [ 1, 3, 17, 14, 8, 38, 9, 41, 19, 48, 22, 6, 30, 33, 43, 11, 46,
    40, 24, 27, 25, 35, 16, 32 ],
  [ 2, 5, 12, 7, 36, 10, 47, 4, 28, 45, 34, 13, 29, 44, 20, 42, 26,
    21, 37, 15, 31, 18, 23, 39 ] ]
```

The first orbit contains the points at the corners, the second those at the edges; clearly the group cannot move a point at a corner onto a point at an edge.

So to investigate the cube group we first investigate the operation on the corner points. Note that the constructed group that describes this operation will operate on the set $[1..24]$, not on the original set $[1,3,17,14,8,38,9,41,19,48,22,6,30,33,43,11,46,40,24,27,25,35,16,32]$.

```
gap> cubel := Action( cube, orbits[1] );
<permutation group with 6 generators>
gap> NrMovedPoints( cubel );
24
gap> Size( cubel );
88179840
```

Now this group obviously operates transitively, but let us test whether it is also primitive.

```
gap> corners := Blocks( cubel, MovedPoints( cubel ) );
[ [ 1, 7, 22 ], [ 2, 14, 20 ], [ 3, 12, 16 ], [ 4, 17, 18 ],
  [ 5, 9, 21 ], [ 6, 10, 24 ], [ 8, 11, 23 ], [ 13, 15, 19 ] ]
```

Those eight blocks correspond to the eight corners of the cube; on the one hand the group permutes those and on the other hand it permutes the three points at each corner cyclically.

So the obvious thing to do is to investigate the operation of the group on the eight corners. The action gives a homomorphism to a permutation group on the corners:

```
gap> blockhom1 := ActionHomomorphism( cubel, corners, OnSets );
<action homomorphism>
```

```

gap> cube1b := Image( blockhom1 );
Group([ (1,2,4,3), (1,3,6,5), (1,5,8,2), (3,4,7,6), (5,6,7,8),
        (2,8,7,4) ])
gap> Size( cube1b );
40320

```

Now a permutation group of degree 8 that has order 40320 must be the full symmetric group $S(8)$ on eight points.

The next thing then is to investigate the kernel of this operation on blocks, i.e., the subgroup of `cube1` of those elements that fix the blocks setwise.

```

gap> Factors( Size( Kernel( blockhom1 ) ) );
[ 3, 3, 3, 3, 3, 3, 3 ]
gap> IsElementaryAbelian( Kernel( blockhom1 ) );
true

```

We can show that the product of this elementary abelian group 3^7 with the $S(8)$ is semidirect by finding a complement, i.e., a subgroup that has trivial intersection with the kernel and that generates `cube1` together with the kernel.

```

gap> cmpl1 := Complementclasses( cube1, Kernel( blockhom1 ) );
[ Group([ (1,3,4,2)(7,16,17,14)(12,18,20,22),
          (1,2,3,4,5,6,13)(7,14,16,17,21,10,15)(9,24,19,22,20,12,18),
          (1,2,3,4,5,8,13)(7,14,16,17,21,23,15)(9,11,19,22,20,12,18) ]) ]
gap> cmpl1 := cmpl1[1];;
gap> Size( cmpl1 );
40320

```

We verify the complement properties:

```

gap> Size( Intersection( cmpl1, Kernel( blockhom1 ) ) );
1
gap> ClosureGroup( cmpl1, Kernel( blockhom1 ) ) = cube1;
true

```

There is even a more elegant way to show that `cmpl1` is a complement.

```

gap> IsBijective( RestrictedMapping( blockhom1, cmpl1 ) );
true

```

Of course, theoretically it is clear that `cmpl1` must indeed be a complement.

In fact we know that `cube1` is a subgroup of index 3 in the wreath product of a cyclic 3 with $S(8)$. This missing index 3 tells us that we do not have total freedom in turning the corners. The following tests show that whenever we turn one corner clockwise we must turn another corner counterclockwise.

```

gap> (1,7,22) in cube1;
false
gap> (1,7,22)(2,20,14) in cube1;
true

```

More or less the same things happen when we consider the operation of the cube group on the edges.

```

gap> cube2 := Action( cube, orbits[2] );;
gap> Size( cube2 );
980995276800
gap> edges := Blocks( cube2, MovedPoints( cube2 ) );

```

```

[ [ 1, 11 ], [ 2, 17 ], [ 3, 19 ], [ 4, 22 ], [ 5, 13 ], [ 6, 8 ],
  [ 7, 24 ], [ 9, 18 ], [ 10, 21 ], [ 12, 15 ], [ 14, 20 ],
  [ 16, 23 ] ]
gap> blockhom2 := ActionHomomorphism( cube2, edges, OnSets );;
gap> cube2b := Image( blockhom2 );;
gap> Size( cube2b );
479001600
gap> Factors( Size( Kernel( blockhom2 ) ) );
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
gap> IsElementaryAbelian( Kernel( blockhom2 ) );
true
gap> cml2 := Complementclasses( cube2, Kernel( blockhom2 ) );;
gap> Length( cml2 );
4

```

So there are even 4 classes of complements here. This time we get a semidirect product of a 2^{11} with an $S(12)$, namely a subgroup of index 2 of the wreath product of a cyclic 2 with $S(12)$. Here the missing index 2 tells us again that we do not have total freedom in turning the edges. The following tests show that whenever we flip one edge we must also flip another edge.

```

gap> (1,11) in cube2;
false
gap> (1,11)(2,17) in cube2;
true

```

Since cube1 and cube2 are the groups describing the actions on the two orbits of cube , it is clear that cube is a subdirect product of those groups, i.e., a subgroup of the direct product. Comparing the sizes of cube1 , cube2 , and cube we see that cube must be a subgroup of index 2 in the direct product of those two groups.

```

gap> Size( cube );
43252003274489856000
gap> Size( cube1 ) * Size( cube2 );
86504006548979712000

```

This final missing index 2 tells us that we cannot operate on corners and edges totally independently. The following tests show that whenever we exchange a pair of corners we must also exchange a pair of edges (and vice versa).

```

gap> (17,19)(11,8)(6,25) in cube;
false
gap> (7,28)(18,21) in cube;
false
gap> (17,19)(11,8)(6,25)(7,28)(18,21) in cube;
true

```

As a last part of the structure analysis of the cube group let us compute the centre of the cube group, i.e., the subgroup of those operations that can be performed either before or after any other operation with the same result.

```

gap> z := Centre( cube );
Group(
[ (2,34)(4,10)(5,26)(7,18)(12,37)(13,20)(15,44)(21,28)(23,42)(29,
  36)(31,45)(39,47) ] )

```

We see that the centre contains one nontrivial element, namely the operation that flips all 12 edges simultaneously.

Finally we turn to the original idea connected with the cube , namely to find a

sequence of turns of the faces that will transform the cube back into its original state. This amounts to a decomposition of a given element of the cube group into a product of the generators. For this purpose we introduce a free group and a homomorphism of it onto the cube group.

```
gap> f := FreeGroup("t", "l", "f", "r", "e", "b");
<free group on the generators [ t, l, f, r, e, b ]>
gap> hom := GroupHomomorphismByImages( f, cube, GeneratorsOfGroup(f),
> GeneratorsOfGroup(cube) );
[ t, l, f, r, e, b ] ->
[ (1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19),
(1,17,41,40)(4,20,44,37)(6,22,46,35)(9,11,16,14)(10,13,15,12),
(6,25,43,16)(7,28,42,13)(8,30,41,11)(17,19,24,22)(18,21,23,20),
(3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28),
(1,14,48,27)(2,12,47,29)(3,9,46,32)(33,35,40,38)(34,37,39,36),
(14,22,30,38)(15,23,31,39)(16,24,32,40)(41,43,48,46)(42,45,47,44) ]
```

Using this homomorphism, we can now decompose elements into generators. The method used utilizes a stabilizer chain and does not enumerate all group elements, therefore the words obtained are not the shortest possible, though they are short enough for hand solutions.

First we decompose the centre element:

```
gap> PreImagesRepresentative( hom, z.1 );
l^-1*t^-1*e^-1*t^2*e*l*f*r*t^-1*r^-1*f^-1*t*f*r*t*r^-1*t^-1*f^-1*t^-1
-f*t*l*t*l^-1*f^-1*l*t^-1*l^-1*f*r*t^-1*r^-1*f^-1*l*t*f*t^-1*f^-1
-l^-1*t*f^-1*l^-1*t^-1*l*t*f*t^-2*f*t*f^-1*t^-1*l^-1*t^-1*l*t^-1*l^-1
-t*e^-1*t*e*l*t*l*t*e*l^-1*e^-1*t^-1*l^-3*b*f*b^-1*l^-1*f^-1*t*l^-1
-f*t*f*l^-1*t^-1*b*r^-1*b^-1*t^-2*e^-1*r*e*r*f^-1*e*t^-1*e^-1*r^-1
-2*t^-2*l^-1*b^-1*r^-1*e^-1
gap> Length( last );
106
```

Next we decompose some element arbitrarily chosen by us:

```
gap> PreImagesRepresentative( hom, (17,19)(11,8)(6,25)(7,28)(18,21) );
l^-1*t^-1*l*f*r*t*r^-1*f^-1*l*t*f*t^-1*f^-1*l^-1*t^2*f*t*l*t*l^-1*f^-1
-l*t^-1*l^-1*f*t^-1*f^-1*l*t*l^-1*t*l*t^-2*l^-1*f*t*r*t^-1*r^-1
-t*r*t^-1*r^-1*f^-1*t*l*f^-1*l^-1*f*l^-1*t^-1*l*t^-2*f*t*f^-1*l^-1
-f^-1*l^-2*f*l*e^-1*t*e*l*t^-1*e^-1*t^-1*e*l*b*f^-1*b^-1
gap> Length( last );
77
```

Last we let GAP choose a random element ...

```
gap> r := Random( cube );
(1,43,6,27,32,46)(2,4,13,34,10,20)(3,38,40,9,24,17)(5,15,45,23,29,47,
26,44,31,42,36,39)(7,18)(8,22)(11,33,48,14,35,30)(12,21,37,28)(16,
19)(25,41)
gap> pre := PreImagesRepresentative( hom, r );
e^-1*r*f^-1*t^-1*r*l*e*b^-1*e^-1*t^-1*b^-1*e^-1*b*l^-1*f^-1*l*f^-1
-l*t^-1*f*b*f^2*b^-1*t*f^-1*l*f*l*t*l^-1*t^2*l^-1*f^-1*l*f*l*f*t^-1
-f^-1*l^-1*t^-1*l*t^2*l^-1*t*l*f^-1*l*f*l^-1*t^-2*l^-1*t^-1*l^-1
-e*l*e^-1*t*l*t^-3*l*t*f*t^-1*f^-2*l^-1*f*t*f*t^-1*f^2*l*f*l^-1*t^-1
-l^-1*t*l*t*f*r*t*r^-1*t^-1*f^-1*t^-1*l^-1*t^-1*e^-1*t*e*l
gap> Length( last );
100
```

... and we verify that the decomposition is correct:

```
gap> Image( hom, pre );  
(1,43,6,27,32,46)(2,4,13,34,10,20)(3,38,40,9,24,17)(5,15,45,23,29,47,  
26,44,31,42,36,39)(7,18)(8,22)(11,33,48,14,35,30)(12,21,37,28)(16,  
19)(25,41)  
gap> last = r;  
true
```

This concludes our example. Of course, GAP can do much more, but demonstrating them all would take too much room.