Solutions for Workshop

Programming and Groups

Alice C. Niemeyer

1. ```
   gap> m11 := MathieuGroup(11);
   Group([ (1,2,3,4,5,6,7,8,9,10,11), (3,7,11,8)(4,10,5,6) ])
   gap> elm11 := Elements( m11);;
   gap> found := false;; gap> for e in elm11 do
   > if Order(e) = 10 then
   > found := true;
   > fi;
   > od;
   gap> found;
   false
   ```
   There is no element of order 10 in M_{11}.

2. ```
   gap> ord5 := Filtered( elm11, e -> Order(e) = 5 ) ;;
   Length(ord5);
   1584
   ```

3. ```
   gap> Filtered( ord5, e -> (2^e = 5 and 5^e = 8 ) );
   [ (2,5,8,11,7)(3,9,6,4,10), (2,5,8,9,10)(3,11,4,6,7),
   (1,2,5,8,3)(4,9,6,7,10), (1,2,5,8,11)(3,7,9,10,6),
   (1,3,2,5,8)(4,10,6,9,11), (1,3,4,7,11)(2,5,8,9,6),
   (1,3,11,7,9)(2,5,8,10,4), (1,6,9,10,4)(2,5,8,7,11),
   (1,6,10,3,11)(2,5,8,4,9), (1,7,11,6,3)(2,5,8,4,10),
   (1,7,10,9,11)(2,5,8,6,4), (1,7,2,5,8)(3,4,9,10,11),
   (1,7,3,9,4)(2,5,8,10,6), (1,10,11,7,3)(2,5,8,6,9),
   (1,10,6,4,7)(2,5,8,3,11), (1,11,6,4,9)(2,5,8,7,3) ]
   ```

4. ```
   gap> NrOrd := function( grp )
   local ele, ords, g, freq, i;
   ele := Elements(grp);
   ```

```
ords := [1];
freq := [0];
for g in ele do
    i := Position( ords, Order(g) );
    if i = fail then
        Add( ords, Order(g) );
        Add( freq, 1 );
    else
        freq[i] := freq[i]+1;
    fi;
od;
return [ords, freq];
end;
```

```
gap> NrOrd( m11 );
[ [ 1, 4, 2, 3, 8, 5, 6, 11 ],
  [ 1, 990, 165, 440, 1980, 1584, 1320, 1440 ] ]
```

Note that another way of doing that is to use `Collect()` to generate a list of order-frequency pairs.

```
gap> Collected( List( elm11, e -> Order( e ) ) );
[ [ 1, 1 ], [ 2, 165 ], [ 3, 440 ], [ 4, 990 ], [ 5, 1584 ],
[ 6, 1320 ], [ 8, 1980 ], [ 11, 1440 ] ]
gap> NrOrd( DihedralGroup( IsPermGroup, 12 ) );
[ [ 1, 2, 6, 3 ], [ 1, 7, 2, 2 ] ]
gap> NrOrd( SymmetricGroup(6) );
[ [ 1, 2, 3, 4, 6, 5 ], [ 1, 75, 80, 180, 240, 144 ] ]
```

5.
```
gap> NrCycleStr := function( grp, g )
local cs, nrs, i;
if not g in grp then return false; fi;
nrs := 0;
for i in Elements(grp) do
    if CycleStructurePerm(i) = CycleStructurePerm(g) then
        nrs := nrs + 1;
    fi;
od;
return nrs;
end;
```

6. (a) gap> ConjugacyClasses(d12);

[()^G, (2,6)(3,5)^G, (1,2)(3,6)(4,5)^G, (1,2,3,4,5,6)^G, (1,3,5)(2,4,6)^G, (1,4)(2,5)(3,6)^G]

Hence there are elements with the same cycle structure which are not conjugate in the group, namely (1,4)(2,5)(3,6) and (1,2)(3,6)(4,5).

(b) gap> ConjugacyClasses(m11);

[()^G, (4,5,10,8)(6,9,7,11)^G, (4,10)(5,8)(6,7)(9,11)^G, (3,4,10)(5,11,6)(7,9,8)^G, (2,3)(4,5,6,11,10,8,7,9)^G, (2,3)(4,8,6,9,10,5,7,11)^G, (2,3,4,8,7)(5,9,6,10,11)^G, (1,2)(3,4,6,5,7,9)(8,11,10)^G, (1,2,3,4,5,6,7,8,9,10,11)^G, (1,2,3,4,6,10,9,8,5,11,7)^G]

Hence there are elements with the same cycle structure which are not conjugate in the group, namely (1,2,3,4,5,6,7,8,9,10,11) and (1,2,3,4,6,10,9,8,5,11,7).

(c) gap> ConjugacyClasses(SymmetricGroup(6));

[()^G, (1,2)^G, (1,2)(3,4)^G, (1,2)(3,4)(5,6)^G, (1,2,3)^G, (1,2,3)(4,5)^G, (1,2,3)(4,5,6)^G, (1,2,3,4)^G, (1,2,3,4)(5,6)^G, (1,2,3,4,5)^G, (1,2,3,4,5,6)^G]

Hence all elements of the same cycle structure are conjugate in S_6.

7. (a)
```
gap> grp := Group( ( 1, 2)(3, 4, 5, 6 ) );
Group([ (1,2)(3,4,5,6) ])
gap> s1 := Stabilizer( grp, 1 );
Group([ (3,5)(4,6) ])
gap> s2 := Stabilizer( s1, 3 );
Group(())
gap> Stabilizer( grp, 3 );
Group(())
```
Hence this group has a base $[1,3]$. Note that $[3]$ is also a base.

(b)
```
gap> grp2 := DihedralGroup( IsPermGroup, 14 );
Group([ (1,2,3,4,5,6,7), (2,7)(3,6)(4,5) ])
gap> s1 := Stabilizer( grp2, 1 );
Group([ (2,7)(3,6)(4,5) ])
gap> s2 := Stabilizer( s1, 2 );
```

```
Group(())
```
Hence D_{14} has a base $[1, 2]$

(c)
```
gap> grp3 := DihedralGroup( IsPermGroup, 16 );
Group([ (1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6) ])
gap> s1 := Stabilizer( grp3, 1 );
Group([ (2,8)(3,7)(4,6) ])
gap> s2 := Stabilizer( s1, 2 );
Group(())
```
Hence D_{16} has a base $[1, 2]$

(d)
```
gap> s1 := Stabilizer( m11, 1 );
Group([ (3,7,11,8)(4,10,5,6), (2,6)(4,9)(5,11)(8,10) ])
gap> s2 := Stabilizer( s1, 2 );
Group([ (3,7,11,8)(4,10,5,6), (4,7,10,6)(5,9,8,11) ])
gap> s3 := Stabilizer( s2, 3 );
Group([ (4,7,10,6)(5,9,8,11), (4,8,10,5)(6,11,7,9) ])
gap> s4 := Stabilizer( s3, 4);
Group(())
```
Hence M_{11} has a base $[1, 2, 3, 4]$

(e)
```
gap> sn := SymmetricGroup( 5 );
Sym( [ 1 .. 5 ] )
gap> s1 := Stabilizer(sn,1);
Group([ (2,5), (3,5), (4,5) ])
gap> s2 := Stabilizer(s1,2);
Group([ (3,5), (4,5) ])
gap> s3 := Stabilizer(s2,3);
Group([ (4,5) ])
gap> s4 := Stabilizer(s3,4);
Group(())
```
Hence S_5 has a base $[1, 2, 3, 4]$

(f)
```
gap> sn := SymmetricGroup( 6 );
Sym( [ 1 .. 6 ] )
gap> s1 := Stabilizer(sn,1);
Group([ (2,6), (3,6), (4,6), (5,6) ])
gap> s2 := Stabilizer(s1,2);
Group([ (3,6), (4,6), (5,6) ])
gap> s3 := Stabilizer(s2,3);
```

```
Group([ (4,6), (5,6) ])
gap> s4 := Stabilizer(s3,4);
Group([ (5,6) ])
gap> s5 := Stabilizer(s4,5);
Group(())
```

Hence S_6 has a base $[1, 2, 3, 4, 5]$

8.
```
gap> Orbit( m11, 1 );
[ 1, 11, 10, 7, 9, 4, 6, 3, 8, 5, 2 ]
gap> s1 := Stabilizer(m11,1);;
gap> Orbit( s1, 2 );
[ 2, 6, 5, 10, 11, 4, 8, 7, 9, 3 ]
gap> s2 := Stabilizer(s1,2);;
gap> Orbit( s2, 3 );
[ 3, 8, 11, 7, 9, 4, 5, 6, 10 ]
gap> s3 := Stabilizer(s2,3);;
gap> Orbit( s3, 4 );
[ 4, 6, 10, 7, 5, 9, 8, 11 ]
```

   Hence the order of M_{11} is $8 \cdot 9 \cdot 10 \cdot 11 = 7920$.

```
gap> Size(m11);
7920
```

9.
```
gap> g := (1,5)(2,11,10,6,9,4,3,7,8);
gap> g in m11;
false
```

10.
```
gap> Factorization( m11, (1,10,4,3,6,9)(2,8)(5,7,11));
x1*x2*x1*x2*x1^-3*x2
gap> m11.1 *m11.2 * m11.1 * m11.2 * m11.1^-3 * m11.2;
(1,10,4,3,6,9)(2,8)(5,7,11)
```