Permutation Groups

May 4, 2022

Contents

Preface

This is my CATAM project, 16.5 for part II. The code for each question can be found in section 5.

1 Permutations

1.1 Question 1

See below tables for outputs of my programs to calculate the inverse and product of permutations of $\{1, \ldots, n\}$. Note permutations are given as column vectors where the vector $(a_1 \ \ldots \ a_n)^T$ corresponds to the permutation $i \mapsto a_i$.

π_1	π_2	$\pi_1 \circ \pi_2$
$\begin{pmatrix}1\\3\\4\\5\\2\end{pmatrix}\begin{pmatrix}2\\3\\1\end{pmatrix}$	$\begin{pmatrix}2\\4\\3\\5\\1\end{pmatrix}\begin{pmatrix}1\\3\\2\end{pmatrix}$	$\begin{pmatrix}3\\5\\4\\2\\1\end{pmatrix}\begin{pmatrix}2\\1\\3\end{pmatrix}$

Table 1: Products of Permutations

π	π^{-1}
$\begin{pmatrix}1\\5\\4\\2\\6\\3\end{pmatrix}\begin{pmatrix}2\\4\\3\\1\end{pmatrix}$	$\begin{pmatrix}1\\4\\6\\3\\2\\5\end{pmatrix}\begin{pmatrix}4\\1\\3\\2\end{pmatrix}$

Table 2: Inverses of Permutations

The complexity of the program to calculate inverses is $O(n^2)$ while the complexity of the program to calculate products is $O(n)$.

2 Groups

2.1 Question 2

Let $\tilde{G}$ denote the group generated by the permutations outputted by the Stripping Algorithm of Sims. Firstly we have $\tilde{G} \leq G$ since all elements in $\tilde{G}$ are generated by elements in G. Then given any $g \in G$, there must exist $g_1, \ldots g_l \in \tilde{G}$ such that

$$g_l^{-1} \cdots g_1^{-1} g = e \text{ or } g' \in \tilde{G}.$$

In either case we have g is equal to a product of elements in $\tilde{G}$ and so itself is in $\tilde{G}$. Hence we have $G = \tilde{G}$.

Given $G = \langle g_1, \ldots, g_k \rangle$, we must have $|\tilde{G}| \leq k$. This is because at most each g_k corresponds to a single element in $\tilde{G}$. The complexity of the algorithm is $O(n^5)$.

2.2 Question 3

See below a table of outputs for the Stripping Algorithm of Sims program. Note the input given as a matrix with each column corresponding to a generator. Further each column vector is written as the permutation

sending $i \mapsto v_i$ with v_i the i^{th} entry of the column v.

Original Generators	Reduced Set of Generators
$\begin{pmatrix} 3 & 4 & 2 & 2 & 4 \\ 1 & 1 & 4 & 3 & 2 \\ 4 & 2 & 3 & 4 & 1 \\ 2 & 3 & 1 & 1 & 3 \end{pmatrix}$ $\begin{pmatrix} 3 & 1 & 3 \\ 2 & 2 & 1 \\ 1 & 3 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 1 & 2 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{pmatrix}$

Table 3: Output of Stripping Algorithm

3 Orbit and Stabililzer

3.1 Question 4

For G acting on a set Ω and $\alpha \in \Omega$, there is a bijection between G/G_α and $G\alpha$ as follows

$$G/G_\alpha \to G\alpha$$
$$gG\alpha \mapsto g\alpha$$

The orbit stabiliser theorem is the above statement. For finite groups there is a corollary of this theorem which is often referred to as the Orbit-Stabiliser Theorem. Under the same set-up as the previously given bijection, this corollary states

$$|G\alpha| = \frac{|G|}{|G_\alpha|}.$$

3.2 Question 5 - add tables

See below for a table of my program which calculates the orbit of an element, along with corresponding witnesses, under the action of permutation group. Note that the orbit of a is given as a row vector, with the entries being the orbit of a. The witnesses are given as a matrix with the i^{th} column being the witness of the i^{th} entry in the orbit vector. The format of the permutation column vectors remains the same as in previous questions.

Generators	Element	Orbit	Witness
$\begin{pmatrix} 3 & 1 & 4 & 4 \\ 1 & 3 & 1 & 1 \\ 4 & 4 & 1 & 2 \\ 2 & 2 & 3 & 3 \end{pmatrix}$ $\begin{pmatrix} 4 & 1 \\ 2 & 2 \\ 5 & 5 \\ 1 & 4 \\ 3 & 3 \end{pmatrix}$	3	$\begin{pmatrix} 3 & 1 & 4 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 4 & 1 & 4 \\ 2 & 3 & 3 & 1 \\ 3 & 1 & 4 & 2 \\ 4 & 2 & 2 & 3 \end{pmatrix}$
	1	$\begin{pmatrix} 1 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 4 \\ 2 & 2 \\ 3 & 5 \\ 4 & 1 \\ 5 & 3 \end{pmatrix}$

Table 4: Orbit of an Element Under a Permutation Group

Let $G = \langle g_1, \ldots, g_k \rangle$. My program works by noting that any element in the orbit of α will be given by a finite combination of g_i applied α in some order. Hence by calculating the image of α under each g_i and recording some g_i gives rise to each image, then iterating that process for each new element in the image

will give us the orbit of x along with witnesses. Further the program will terminate as by finiteness of Ω, the set G acts on, we will eventually reach every element in the orbit. In which case applying each g_i to this set will give us no new elements and so the program will terminate.

4 Schreier's Theorem and the Final Algorithm

4.1 Question 6

Note that by the definition of t_i we have

$$\begin{aligned} t_1 G_\alpha &= G_\alpha \\ t_{i+1} G_\alpha &= y_i t_i G_\alpha. \end{aligned}$$

Hence we have

$$\begin{aligned} t_{r+1} G_\alpha &= y_r \cdots y_1 t_1 G_\alpha \\ &= x t_1 G_\alpha \\ &= G_\alpha. \end{aligned}$$

Further, we have $t_{r+1} \in \mathrm{im}(\varphi) = T$ and so $t_{r+1} = t_1$. Then for any $x \in G_\alpha$ we have

$$\begin{aligned} x &= y_r \cdots y_1 \\ &= t_{r+1}\big(t_{r+1}^{-1} y_r t_r\big) \cdots \big(t_2^{-1} y_1 t_1\big) t_1^{-1} \\ &= t_1\big(\varphi(y_r t_r)^{-1} y_r t_r\big) \cdots \big(\varphi(y_1 t_1)^{-1} y_1 t_1\big) t_1^{-1}. \end{aligned}$$

Letting $\Lambda = \big\langle \varphi(yt)^{-1} yt : y \in Y, t \in T \big\rangle$ and applying this to $x = t_1$ gives $t_1 \in \Lambda$. Hence for any $x \in G_\alpha$ we have

$$x \in t_1 \Lambda t_1^{-1} \subset \Lambda.$$

It follows that $G_\alpha \subset \Lambda$. Then $(\forall y \in Y)(\forall t \in T)$ we have

$$\begin{aligned} \varphi(yt) G_\alpha &= yt G_\alpha \\ \Rightarrow \varphi(yt)^{-1} yt &\in G_\alpha. \end{aligned}$$

Then the reverse inclusion follows and we must have $G_\alpha = \Lambda$.

4.2 Question 7 - comment on complexity

See below for a table of a set of generators of the stabiliser of an element of a permutation group.

Generator	Element	Stabiliser
$\begin{pmatrix} 2 & 2 \\ 1 & 4 \\ 3 & 3 \\ 4 & 1 \end{pmatrix}$	2	$\begin{pmatrix} 4 \\ 2 \\ 3 \\ 1 \end{pmatrix}$
$\begin{pmatrix} 2 & 5 & 1 \\ 4 & 4 & 4 \\ 5 & 1 & 2 \\ 1 & 2 & 5 \\ 3 & 3 & 3 \end{pmatrix}$	4	$\begin{pmatrix} 2 & 3 & 1 & 1 & 1 \\ 3 & 1 & 3 & 5 & 2 \\ 5 & 2 & 5 & 2 & 5 \\ 4 & 4 & 4 & 4 & 4 \\ 1 & 5 & 2 & 3 & 3 \end{pmatrix}$

Table 5: Stabiliser of an Element Under a Permutation Group

4.3 Question 8 - add tables

See below for a table of the size of the group generated by a given set of permutations.

Generators	Group Size
$\begin{pmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{pmatrix}$	6
$\begin{pmatrix} 1 & 1 & 4 \\ 3 & 3 & 3 \\ 4 & 2 & 1 \\ 2 & 4 & 2 \end{pmatrix}$	24

Table 6: Size of Permutation Group Generated by Given Set of Permutations

If we didn't run the stripping algorithm at every stage in our program, the efficiency of the program would be greatly reduced. This is because at each stage we would (in most cases) be increasing our number of generators for the group. This would then lead to an increase in time to calculate the orbit and witnesses. This would then further increase the time to calculate the stabiliser and further the number of generators of the stabiliser. This would then spill over into the next repeat of the algorithm, exponentially increasing the number of calculations and storage required for the program to run. For $n = 20$ this would be a massive increase in time to run the program.

4.4 Question 9 - add tables

From IA, we know that $S_n = \langle (1\ 2), (1\ 2\ \ldots\ n) \rangle$, hence for all n we must have $P_n > 0$. Further note that $A_n \leq S_n$ and

$$|A_n| = \frac{|S_n|}{2},$$

so the probability of choosing two elements at random from A_n is $\frac{1}{4}$, hence we have for all n that $P_n \leq \frac{3}{4}$. Since if we choose both elements from A_n then the subgroup the generate is contained in A_n which is strictly contained in S_n.

Since $S_2 = C_2$, $P_2 = \frac{3}{4}$. Then for $n = 3$ we have that S_3 is generated by any 3 and 2 cycle, any two 2 cycles and these are precisely the combinations of two permutations which work. To prove this, first note for any $a \neq b \neq c \neq a$ $(a\ b\ c), (a\ b)$ generate S_3 as previously discussed. Then we have

$(a\ b\ c)(a\ c) = (b\ c)$ and so S_3 is generated by any 2 and 3 cycle,
$(a\ b)(b\ c) = (a\ b\ c)$ and so any two different 2 cycles generate S_3,
$(a\ b\ c)(a\ c\ b) = (a\ b\ c)(a\ b\ c) = e$ and so any two 3 cycles don't generate S_3.

Further S_3 can't be generated by any single permutation and so we have the desired result. By counting the possible generators, it then follows

$$P_3 = \frac{18}{36} = \frac{1}{2}.$$

My program generates random permutations by setting $\sigma(1) =$ a random number in $\{1, \ldots, n\}$. Then iteratively generating a new random integer l in $\{1, \ldots, n\}$ and if it is not equal to any previous $\sigma(i')$ then set $\sigma(i) = l$.

See below for my estimates of P_n for $5 \leq n \leq 10$.

n	5	6	7	8	9	10
P_n	0.35	0.38	0.54	0.56	0.63	0.63

Table 7: Values of P_n for various n

5 Code

5.1 Question 1

```
function [outputArg1,outputArg2] = timespi(pia,pib)
n=size(pia,1);


i=1;

pipiab=zeros(n,1);

while i<= n
    pipiab(i)=pia(pib(i));
    i=i+1;
end
pipiab


end
```

```
function [outputArg1,outputArg2] = invpi(pi)
n=size(pi,1);

i=1;

inv=zeros(n,1);
while i<= n
    term=0;
    j=1;
    while term == 0

    if pi(j)==i;
        inv(i)=j;
        term=1;
    else
        j=j+1;
    end
    end
    i=i+1;
end
inv


end
```

5.2 Question 3

```
function [outputArg1,outputArg2] = stripping(Pi)

%input(pi_1,pi_2, ... ,pi_k)
%Going to have to change input to a matrix with each collumn being a
   generator
```

```
5
6   k=size(Pi,2);
7
8   n=size(Pi,1);
9
10  l=1;
11
12  A=zeros(n,n);
13
14  while l <= k
15
16          r=1;
17          finish=0;
18
19          while finish == 0
20
21                  if r~=n
22
23                          if Pi(r,l) ~= r
24
25                                  if A(r,Pi(r,l)) == 0
26
27                                          A(r,Pi(r,l))=l;
28                                          finish=1;
29
30                                  else
31
32                                          pi=Pi(:,A(r,Pi(r,l)));
33                                          %"run invert pi"
34                          i=1;
35
36                              inv=zeros(n,1);
37                              while i<= n
38                                  term=0;
39                                  j=1;
40                                  while term == 0
41
42                                  if pi(j)==i;
43                                      inv(i)=j;
44                                      term=1;
45                                  else
46                                      j=j+1;
47                                  end
48                                  end
49                                  i=i+1;
50                              end
51
52
53                                          pia=inv;
54                                          pib=Pi(:,l);
55                                          %"run multiply pia,pib program"
56
57                          i=1;
58
```

```
                        pipiab=zeros(n,1);

                        while i<= n
                            pipiab(i)=pia(pib(i));
                            i=i+1;
                        end
                       Pi(:,l)= pipiab;


                    %Ran multiply pia,pib program
                                        r=r+1;

                            end

                        else

                            r=r+1;

                        end

                else

                        finish = 1;

                end

        end
        l=l+1;

end
A;
i=1;
kk=1;
Qi=zeros(n,1);
while i<= size(A,1)
    j=1;
    while j<= size(A,2)
        if A(i,j) ~= 0
        Qi(:, kk)=Pi(:,A(i,j));
        kk=kk+1;
        else
        end
        j=j+1;
    end
    i=i+1;
end
Pi=Qi;
Pi;
A
```

```
end
```

5.3 Question 5

```
function [outputArg1,outputArg2] = orbit(A,el)

k=size(A,2);
n=size(A,1);
orbit=[el,0];
newsize=1;
oldsize=0;

while oldsize < newsize



    i=oldsize+1;
    oldsize=size(orbit,1);
    c=1;

    while i <= oldsize

        j=1;
        while j<= k
```

```
          j;

            if ismembertol(A(orbit(i,1),j),orbit(:,1))

            else
                orbit;
                holdvec=[orbit(i,:),j];
                orbit(:,size(orbit,2)+1)=zeros(size(orbit,1),1);
                orbit(oldsize+c,:)=holdvec;
                orbit(oldsize+c,1)=A(orbit(i,1),j);
                c=c+1;
            end
            j=j+1;
        end
        i=i+1;
    end
newsize=size(orbit,1);
end
orbit


%Calculate table of witnesses
xx=size(orbit,2);
orbsize=size(orbit,1);
i=2;
witness=[1:n].';
B=zeros(size(A,1),size(A,2)+1);
B(:,1)=[1:n];
B(:,[2:size(A,2)+1])=A;

while i<=orbsize

    j=2;
    witness(:,i)=B(:,1);
    while j<= xx
        %"do witness(:,i)=B(:,j+1)*witness(:,i)

        pia=B(:,orbit(i,j)+1);
        pib=witness(:,i);
        pipiab=zeros(n,1);
        ii=1;

            while ii<= n
                pipiab(ii)=pia(pib(ii));
                ii=ii+1;
            end
            witness(:,i)=pipiab;
        j=j+1;
    end
    i=i+1;
end
witness
```

```
end
```

5.4 Question 7

```
function [outputArg1,outputArg2] = stabiliser(Y,a)


Pi=Y;
```

```
el=a;

%Run stripping algorithm"

                  k=size(Pi,2);

                  n=size(Pi,1);

                  l=1;

                  A=zeros(n,n);

                  while l <= k

                      r=1;
                      finish=0;

                      while finish == 0

                          if r~=n

                              if Pi(r,l) ~= r

                                  if A(r,Pi(r,l)) == 0

                                      A(r,Pi(r,l))=l;
                                      finish=1;

                                  else

                                      pi=Pi(:,A(r,Pi(r,l)));
                                      %"run invert pi"
                                          i=1;

                                               inv=zeros(n,1);
                                               while i<= n
                                                   term=0;
                                                   j=1;
                                                   while term == 0

                                                   if pi(j)==i;
                                                       inv(i)=j;
                                                       term=1;
                                                   else
                                                       j=j+1;
                                                   end
                                                   end
                                                   i=i+1;
                                               end


                                      pia=inv;
                                      pib=Pi(:,l);
                                      %"run multiply pia,pib program"
```

```
                                        i=1;

                                        pipiab=zeros(n,1);

                                        while i<= n
                                            pipiab(i)=pia(pib(i));
                                            i=i+1;
                                        end
                                       Pi(:,l)= pipiab;


                                    %Ran multiply pia,pib program
                                    r=r+1;

                                end

                            else

                                r=r+1;

                            end

                        else

                            finish = 1;

                        end

                    end
                    l=l+1;

                end
                A;
                i=1;
                kk=1;
                Qi=zeros(n,1);
                while i<= size(A,1)
                    j=1;
                    while j<= size(A,2)
                        if A(i,j) ~= 0
                        Qi(:, kk)=Pi(:,A(i,j));
                        kk=kk+1;
                        else
                        end
                        j=j+1;
                    end
                    i=i+1;
                end
                Pi=Qi;


%Ran stripping algorithm
Y=Pi;
```

```
y=size(Y,2);

A=Y;
%Run orbit algorithm;

            k=size(A,2);
            n=size(A,1);
            orbit=[el,0];
            newsize=1;
            oldsize=0;

            while oldsize < newsize



                i=oldsize+1;
                oldsize=size(orbit,1);
                c=1;

                while i <= oldsize

                    j=1;
                    while j<= k
                      j;

                        if ismembertol(A(orbit(i,1),j),orbit(:,1))

                        else
                            orbit;
                            holdvec=[orbit(i,:),j];
                            orbit(:,size(orbit,2)+1)=zeros(size(orbit,1)
                                ,1);
                            orbit(oldsize+c,:)=holdvec;
                            orbit(oldsize+c,1)=A(orbit(i,1),j);
                            c=c+1;
                        end
                        j=j+1;
                    end
                    i=i+1;
                end
            newsize=size(orbit,1);
            end
            orbit;


            %Calculate table of witnesses
            xx=size(orbit,2);
            orbsize=size(orbit,1);
            i=2;
            witness=[1:n].';
            B=zeros(size(A,1),size(A,2)+1);
            B(:,1)=[1:n];
            B(:,[2:size(A,2)+1])=A;

```

```
166             while i<=orbsize
167
168                 j=2;
169                 witness(:,i)=B(:,1);
170                 while j<= xx
171                     %"do witness(:,i)=B(:,j+1)*witness(:,i)
172
173                     pia=B(:,orbit(i,j)+1);
174                     pib=witness(:,i);
175                     pipiab=zeros(n,1);
176                     ii=1;
177
178                         while ii<= n
179                             pipiab(ii)=pia(pib(ii));
180                             ii=ii+1;
181                         end
182                         witness(:,i)=pipiab;
183                     j=j+1;
184                 end
185                 i=i+1;
186             end
187             witness;
188
189
190
191
192
193 %Ran orbit program
194 t=size(orbit,1);
195 T=witness;
196
197
198
199
200
201
202
203  yi=1;
204
205
206  stab=zeros(n,1);
207
208  while yi <= y
209
210         ti=1;
211
212         while ti<= t
213
214
215                 %Calculate Y(:,yi)*T(:,ti)
216
217                         pia=Y(:,yi);
218                         pib=T(:,ti);
219
```

```
                %"Run timespi program"

        i=1;

        pipiab=zeros(n,1);

        while i<= n
            pipiab(i)=pia(pib(i));
            i=i+1;
        end



        %Calculate varphi(piapib)


             ne=pipiab(a);
                              i=1;
                              foundinverse=0;
                              while foundinverse==0
                                  if orbit(i,1)==ne
                                      nu=i;
                                      foundinverse=1;
                                  else
                                      i=i+1;
                                  end
                              end



                              varphi=T(:,nu);

        %Calculate inverse of varphi

                pi=varphi;

                %"run invpi program "

    i=1;

    inv=zeros(n,1);
    while i<= n
        term=0;
        j=1;
        while term == 0

        if pi(j)==i;
            inv(i)=j;
            term=1;
        else
            j=j+1;
        end
        end
        i=i+1;
```

```
            end


                %Calculate stab

                        pia=inv;
                        pib=pipiab;

                        %"Run timespi program"

                    i=1;

                    pipiab=zeros(n,1);

                    while i<= n
                        pipiab(i)=pia(pib(i));
                        i=i+1;
                    end



                        stab(:,(yi-1)*y+ti)=pipiab;




                ti=ti+1;
        end
        yi=yi+1;
 end

stab;
Pi=stab;

%Run stripping algorithm


                k=size(Pi,2);

                n=size(Pi,1);

                l=1;

                A=zeros(n,n);

                while l <= k

                    r=1;
                    finish=0;

                    while finish == 0

                        if r~=n

```

```
                if Pi(r,l) ~= r

                    if A(r,Pi(r,l)) == 0

                        A(r,Pi(r,l))=l;
                        finish=1;

                    else

                        pi=Pi(:,A(r,Pi(r,l)));
                        %"run invert pi"
                            i=1;

                                inv=zeros(n,1);
                                while i<= n
                                    term=0;
                                    j=1;
                                    while term == 0

                                    if pi(j)==i;
                                        inv(i)=j;
                                        term=1;
                                    else
                                        j=j+1;
                                    end
                                    end
                                    i=i+1;
                                end


                        pia=inv;
                        pib=Pi(:,l);
                        %"run multiply pia,pib program"

                            i=1;

                            pipiab=zeros(n,1);

                            while i<= n
                                pipiab(i)=pia(pib(i));
                                i=i+1;
                            end
                           Pi(:,l)= pipiab;


                        %Ran multiply pia,pib program
                        r=r+1;

                    end

                else

                    r=r+1;

```

```
                                end

                            else

                                finish = 1;

                            end

                        end
                        l=l+1;

                    end
                    A;
                    i=1;
                    kk=1;
                    Qi=zeros(n,1);
                    while i<= size(A,1)
                        j=1;
                        while j<= size(A,2)
                            if A(i,j) ~= 0
                            Qi(:, kk)=Pi(:,A(i,j));
                            kk=kk+1;
                            else
                            end
                            j=j+1;
                        end
                        i=i+1;
                    end
                    Pi=Qi;









%Ran stripping algorithm

stab=Pi;

stab









end
```

5.5 Question 8

```
function [outputArg1,outputArg2] = permgroupsize(G)
%Finding the size of G
oldGsize(1)=size(G,2);
iii=1;
nn(iii)=1;
nontrivorb=0;


while nontrivorb == 0


%Step 1: Reduce generators of G
            clear Pi
            Pi=G;
            %Run stripping algorithm
                k=size(Pi,2);

                n=size(Pi,1);

                l=1;

                A=zeros(n,n);

                while l <= k && ~ismember(0,Pi)


                    r=1;
                    finish=0;

                    while finish == 0

                        if r~=n

                            if Pi(r,l) ~= r
                                    Pi(r,l);
                                if A(r,Pi(r,l)) == 0

                                    A(r,Pi(r,l))=l;
                                    finish=1;

                                else

                                    pi=Pi(:,A(r,Pi(r,l)));
                                    %"run invert pi"
                                        i=1;

                                            inv=zeros(n,1);
                                            while i<= n
                                                term=0;
                                                j=1;
                                                while term == 0

```

```
                                        if pi(j)==i;
                                            inv(i)=j;
                                            term=1;
                                        else
                                            j=j+1;
                                        end
                                        end
                                        i=i+1;
                                    end


                                pia=inv;
                                pib=Pi(:,l);
                                %"run multiply pia,pib program"

                                    i=1;

                                    pipiab=zeros(n,1);

                                    while i<= n
                                        pipiab(i)=pia(pib(i));
                                        i=i+1;
                                    end
                                    pipiab;
                                   Pi(:,l)= pipiab;


                                %Ran multiply pia,pib program
                                r=r+1;

                            end

                        else

                            r=r+1;

                        end

                    else

                        finish = 1;

                    end

                end
                l=l+1;

            end
            A;
            i=1;
            kk=1;

            Qi=zeros(n,1);
            while i<= size(A,1)
```

```
                    j=1;
                    while j<= size(A,2)
                        if A(i,j) ~= 0
                        Qi(:, kk)=Pi(:,A(i,j));
                        kk=kk+1;
                        else
                        end
                        j=j+1;
                    end
                    i=i+1;
                end
                Qi;
                Pi=Qi;
                G=Pi;


        newGsize(iii)=size(G,2);
        %call reduced generators G too

        if G(1,1)==0
                nontrivorb=1;
        end

        if nontrivorb==0

%Step 2: Find non-trivial orbits of G

        pi=G(:,1);
        i=1;
    foundel=0;
        while foundel==0

                if pi(i)==i
                        i=i+1;
                else
                foundel=1;
                a=i;
                end
        end



                        %Step 3: Find orbit of a

                    %Run stabiliser program

                        Y=G;
                        y=size(Y,2);

                            A=Y;
                            el=a;
                            %Run orbit algorithm;

```

```
k=size(A,2);
n=size(A,1);
orbit=[el,0];
newsize=1;
oldsize=0;

while oldsize < newsize



    i=oldsize+1;
    oldsize=size(orbit,1);
    c=1;

    while i <= oldsize

        j=1;
        while j<= k
          j;

            if ismembertol(A(orbit
               (i,1),j),orbit(:,1)
               )

            else
                orbit;
                holdvec=[orbit(i
                   ,:),j];
                orbit(:,size(orbit
                   ,2)+1)=zeros(
                   size(orbit,1)
                   ,1);
                orbit(oldsize+c,:)
                   =holdvec;
                orbit(oldsize+c,1)
                   =A(orbit(i,1),j
                   );
                c=c+1;
            end
            j=j+1;
        end
        i=i+1;
    end
newsize=size(orbit,1);
end
orbit;


%Calculate table of witnesses
xx=size(orbit,2);
orbsize=size(orbit,1);
i=2;
witness=[1:n].';
B=zeros(size(A,1),size(A,2)+1);
```

```
                    B(:,1)=[1:n];
                    B(:,[2:size(A,2)+1])=A;

                    while i<=orbsize

                        j=2;
                        witness(:,i)=B(:,1);
                        while j<= xx
                            %"do witness(:,i)=B(:,j+1)
                               *witness(:,i)

                            pia=B(:,orbit(i,j)+1);
                            pib=witness(:,i);
                            pipiab=zeros(n,1);
                            ii=1;

                                while ii<= n
                                    pipiab(ii)=pia(pib
                                       (ii));
                                    ii=ii+1;
                                end
                                witness(:,i)=pipiab;
                            j=j+1;
                        end
                        i=i+1;
                    end
                    witness;



%Ran orbit program
t=size(orbit,1);
T=witness;

%Find inverse of orbit
    %pi=orbit(:,1)
    %"Run invpi program



    %i=1;

    %inv=zeros(n,1);
    %while i<= n
       % term=0;
        %j=1;
       % while term == 0

       % if pi(j)==i;
          %  inv(i)=j;
           % term=1;
      %  else
```

```
%    j=j+1;
% end
%  end
%  i=i+1;
% end


% inverseorbit=inv;
%Found inverse orbit



yi=1;

clear stab
stab=zeros(n,1);
stab;

while yi <= y

   ti=1;

   while ti<= t


       %Calculate Y(:,yi)*T(:,ti)

           pia=Y(:,yi);
           pib=T(:,ti);

           %"Run timespi program"

               i=1;

               pipiab=zeros(n,1);

               while i<= n
                   pipiab(i)=pia(pib(i));
                   i=i+1;
               end


       %Calculate varphi(piapib)

           ne=pipiab(a);
           i=1;
           foundinverse=0;
           while foundinverse==0
               if orbit(i,1)==ne
                   nu=i;
                   foundinverse=1;
```

```
312                 else
313                     i=i+1;
314                 end
315             end
316
317
318
319             varphi=T(:,nu);
320
321         %Calculate inverse of varphi
322
323             pi=varphi;
324
325             %"run invpi program "
326
327             i=1;
328
329             inv=zeros(n,1);
330             while i<= n
331                 term=0;
332                 j=1;
333                 while term == 0
334
335                 if pi(j)==i
336                     inv(i)=j;
337                     term=1;
338                 else
339                     j=j+1;
340                 end
341                 end
342                 i=i+1;
343             end
344
345
346         %Calculate stab
347
348             pia=inv;
349             pib=pipiab;
350
351             %"Run timespi program"
352
353                     i=1;
354
355                     pipiab=zeros(n,1);
356
357                     while i<= n
358                         pipiab(i)=pia(pib(i));
359                         i=i+1;
360                     end
361                     pipiab;
362
363
364                         (yi-1)*y+(ti);
365
```

```
                                        stab(:,(yi-1)*y+ti)=pipiab;
                                    ti=ti+1;
                                end
                                yi=yi+1;
                             end

                            stab;
                                nn(iii)=size(orbit,1);
                        G;
                        G=stab;
            G;
             %remove 0 cols of G
                            i=1;
                            width=size(G,2);

                            while i<= width
                                if norm(G(:,i))==0
                                    G=G(:,[[1:i-1],[i+1:width]]);
                                    width=width-1;
                                else
                                    i=i+1;
                                end

                            end

            G;
                        oldGsize(iii+1)=size(G,2);

                        %DONT REDUCE GENERATORS OF G yet
        else
        end
        iii=iii+1;
end
nn;
pgroupsize=prod(nn)


end
```

5.6 Question 9

```
function [outputArg1,outputArg2] = estimateP_n(range)
rangeindex=1;
maximumsizeofrange=size(range,2);
while rangeindex<= maximumsizeofrange
    n=range(rangeindex);
nnnn=1;
prob=0;
oldGsize=0;
newGsize=0;
while nnnn<= 10000
    G=zeros(n,1);
    %Generate G_(:,1)
    i=2;
    G(1,1)=randi(n);

    while i<= n
        x=randi(n);
        if ismembertol(x,G([1:i-1],1))
        else
            G(i,1)=x;
            i=i+1;
        end
    end

    %Generated G(:,1)

    %Generate G_(:,1)
    i=2;
    G(1,2)=randi(n);

    while i<= n
        x=randi(n);
        if ismembertol(x,G([1:i-1],2))
        else
            G(i,2)=x;
            i=i+1;
        end
    end

    %Generated G(:,1)




    G;
    nn=1;
    clear oldGsize newGsize


    %run perm group size program
                oldGsize(1)=size(G,2);
                iii=1;
```

```
53
54                  nn(iii)=1;
55                  nontrivorb=0;
56
57
58                  while nontrivorb == 0
59
60
61                  %Step 1: Reduce generators of G
62                              clear Pi
63                              Pi=G;
64                              %Run stripping algorithm
65                                  k=size(Pi,2);
66
67                                  n=size(Pi,1);
68
69                                  l=1;
70
71                                  A=zeros(n,n);
72
73                                  while l <= k
74
75                                      r=1;
76                                      finish=0;
77
78                                      while finish == 0
79
80                                          if r~=n
81
82                                              if Pi(r,l) ~= r
83                                                      Pi;
84                                                  if A(r,Pi(r,l)) == 0
85
86                                                      A(r,Pi(r,l))=l;
87                                                      finish=1;
88
89                                                  else
90
91                                                      pi=Pi(:,A(r,Pi(r,l)));
92                                                      %"run invert pi"
93                                                          i=1;
94
95                                                              inv=zeros(n,1)
                                                                    ;
96                                                              while i<= n
97                                                                  term=0;
98                                                                  j=1;
99                                                                  while term
                                                                        == 0
100
101                                                                 if pi(j)==
                                                                        i;
102                                                                  inv(i)
                                                                        =j;
```

```
                                                    term
                                                       =1;
                                                else
                                                    j=j+1;
                                                end
                                                end
                                                i=i+1;
                                            end


                                        pia=inv;
                                        pib=Pi(:,l);
                                        %"run multiply pia,pib
                                            program"

                                        i=1;

                                        pipiab=zeros(n,1);

                                        while i<= n
                                            pipiab(i)=pia(
                                               pib(i));
                                            i=i+1;
                                        end
                                       Pi(:,l)= pipiab;


                                        %Ran multiply pia,pib
                                           program
                                        r=r+1;

                                    end

                                else

                                    r=r+1;

                                end

                            else

                                finish = 1;

                            end

                        end
                        l=l+1;

                    end
                    A;
                    i=1;
                    kk=1;
                    Qi=zeros(n,1);
                    while i<= size(A,1)
```

```
                                    j=1;
                                    while j<= size(A,2)
                                        if A(i,j) ~= 0
                                        Qi(:, kk)=Pi(:,A(i,j));
                                        kk=kk+1;
                                        else
                                        end
                                        j=j+1;
                                    end
                                    i=i+1;
                                end
                                Pi=Qi;
                                G=Pi;


            newGsize(iii)=size(G,2);
            %call reduced generators G too

            if G(1,1)==0
                nontrivorb=1;
            end

            if nontrivorb==0

        %Step 2: Find non-trivial orbits of G

            pi=G(:,1);
            i=1;
            foundel=0;
            while foundel==0

                if pi(i)==i
                    i=i+1;
                else
                foundel=1;
                a=i;
                end
            end



                    %Step 3: Find orbit of a

                            %Run stabiliser program

                                Y=G;
                                y=size(Y,2);

                                    A=Y;
                                    el=a;
                                    %Run orbit algorithm;

                                                k=size(A,2);
```

```
207                                                 n=size(A,1);
208                                                 orbit=[el,0];
209                                                 newsize=1;
210                                                 oldsize=0;
211
212                                                 while oldsize <
                                                        newsize
213
214
215
216                                                     i=oldsize+1;
217                                                     oldsize=size(
                                                            orbit,1);
218                                                     c=1;
219
220                                                     while i <=
                                                            oldsize
221
222                                                         j=1;
223                                                         while j<=
                                                                k
224                                                           j;
225
226                                                             if
                                                                    ismembertol
                                                                    (A(
                                                                    orbit
                                                                    (i
                                                                    ,1)
                                                                    ,j)
                                                                    ,
                                                                    orbit
                                                                    (:,1)
                                                                    )
227
228                                                             else
229                                                              orbit
                                                                    ;
230                                                              holdvec
                                                                    =[
                                                                    orbit
                                                                    (
                                                                    i
                                                                    ,:)
                                                                    ,
                                                                    j
                                                                    ];
231                                                              orbit
                                                                    (:,
                                                                    size
                                                                    (
                                                                    orbit
```

```
                                                                    ,2)
                                                                    +1)
                                                                    =
                                                                    zeros
                                                                    (
                                                                    size
                                                                    (
                                                                    orbit
                                                                    ,1)
                                                                    ,1)
                                                                    ;
232                                                             orbit
                                                                    (
                                                                    oldsize
                                                                    +
                                                                    c
                                                                    ,:)
                                                                    =
                                                                    holdvec
                                                                    ;
233                                                             orbit
                                                                    (
                                                                    oldsize
                                                                    +
                                                                    c
                                                                    ,1)
                                                                    =
                                                                    A
                                                                    (
                                                                    orbit
                                                                    (
                                                                    i
                                                                    ,1)
                                                                    ,
                                                                    j
                                                                    )
                                                                    )
                                                                    ;
234                                                             c=
                                                                    c
                                                                    +1;
235                                                                 end
236                                                                 j=j+1;
237                                                         end
238                                                         i=i+1;
239                                                     end
240                                             newsize=size(orbit
                                                    ,1);
241                                             end
242                                             orbit;
243
```

```
244
245                                                     %Calculate table
                                                           of witnesses
246                                                     xx=size(orbit,2);
247                                                     orbsize=size(orbit
                                                           ,1);
248                                                     i=2;
249                                                     witness=[1:n].';
250                                                     B=zeros(size(A,1),
                                                           size(A,2)+1);
251                                                     B(:,1)=[1:n];
252                                                     B(:,[2:size(A,2)
                                                           +1])=A;
253
254                                                     while i<=orbsize
255
256                                                         j=2;
257                                                         witness(:,i)=B
                                                               (:,1);
258                                                         while j<= xx
259                                                             %"do
                                                                   witness
                                                                   (:,i)=B
                                                                   (:,j+1)
                                                                   *
                                                                   witness
                                                                   (:,i)
260
261                                                             pia=B(:,
                                                                   orbit(i
                                                                   ,j)+1);
262                                                             pib=
                                                                   witness
                                                                   (:,i);
263                                                             pipiab=
                                                                   zeros(n
                                                                   ,1);
264                                                             ii=1;
265
266                                                              while
                                                                    ii
                                                                    <=
                                                                    n
267                                                                  pipiab
                                                                       (
                                                                       ii
                                                                       )
                                                                       =
                                                                       pia
                                                                       (
                                                                       pib
                                                                       (
                                                                       ii
                                                                       )
```

```
                                                                    )
                                                                    ;
268                                                                   ii
                                                                      =
                                                                      ii
                                                                      +1;
269                                                                 end
270                                                                 witness
                                                                       (:,
                                                                       i)=
                                                                       pipiab
                                                                       ;
271                                                           j=j+1;
272                                                       end
273                                                       i=i+1;
274                                                 end
275                                                 witness;
276
277
278
279
280
281                                         %Ran orbit program
282                                         t=size(orbit,1);
283                                         T=witness;
284
285                                         %Find inverse of orbit
286                                             %pi=orbit(:,1)
287                                             %"Run invpi program
288
289
290
291                                             %i=1;
292
293                                             %inv=zeros(n,1);
294                                             %while i<= n
295                                                % term=0;
296                                                 %j=1;
297                                                % while term == 0
298
299                                                % if pi(j)==i;
300                                                   %  inv(i)=j;
301                                                    % term=1;
302                                               %  else
303                                                  %   j=j+1;
304                                                % end
305                                               %  end
306                                               %  i=i+1;
307                                           % end
308
309
310
```

```
311                                    % inverseorbit=inv;
312                                 %Found inverse orbit
313
314
315
316
317                                 yi=1;
318
319
320                                 stab=zeros(n,1);
321
322                                 while yi <= y
323
324                                    ti=1;
325
326                                    while ti<= t
327
328
329                                        %Calculate Y(:,yi)*T
                                               (:,ti)
330
331                                         pia=Y(:,yi);
332                                         pib=T(:,ti);
333
334                                         %"Run timespi
                                               program"
335
336                                          i=1;
337
338                                          pipiab=zeros(n
                                                ,1);
339
340                                          while i<= n
341                                              pipiab(i)=
                                                    pia(pib
                                                    (i));
342                                              i=i+1;
343                                          end
344
345
346
347                                        %Calculate varphi(
                                               piapib)
348
349                                         ne=pipiab(a);
350                                         i=1;
351                                         foundinverse=0;
352                                         while foundinverse
                                               ==0
353                                          if orbit(i,1)
                                                ==ne
354                                              nu=i;
355                                              foundinverse
                                                    =1;
```

```
356                                                 else
357                                                     i=i+1;
358                                                 end
359                                             end
360
361
362
363                                             varphi=T(:,nu);
364
365                                         %Calculate inverse of
                                                varphi
366
367                                             pi=varphi;
368
369                                             %"run invpi
                                                    program "
370
371                                             i=1;
372
373                                             inv=zeros(n,1);
374                                             while i<= n
375                                                 term=0;
376                                                 j=1;
377                                                 while term ==
                                                        0
378
379                                                 if pi(j)==i
380                                                     inv(i)=j;
381                                                     term=1;
382                                                 else
383                                                     j=j+1;
384                                                 end
385                                                 end
386                                                 i=i+1;
387                                             end
388
389
390                                         %Calculate stab
391
392                                             pia=inv;
393                                             pib=pipiab;
394
395                                             %"Run timespi
                                                    program"
396
397                                                     i=1;
398
399                                                     pipiab=
                                                            zeros(n
                                                            ,1);
400
401                                                     while i<=
                                                            n
```

```
                                                                    pipiab
                                                                        (i)
                                                                        =
                                                                        pia
                                                                        (
                                                                        pib
                                                                        (i)
                                                                        );
                                                                    i=i+1;
                                                                end



                                                        stab(:,(yi-1)*y+ti
                                                            )=pipiab;



                                                            ti=ti+1;
                                                        end
                                                        yi=yi+1;
                                                    end

                                                    stab;



                                    %Found orbit of a



                                        nn(iii)=size(orbit,1);



                                    G=stab;

                                    %remove 0 cols of G
                                    i=1;
                                    width=size(G,2);

                                    while i<= width
                                        if norm(G(:,i))==0
                                            G=G(:,[[1:i-1],[i+1:width]]);
                                            width=width-1;
                                        else
                                            i=i+1;
                                        end

                                    end
                                    G;
                                    oldGsize(iii+1)=size(G,2);
```

```
                                %DONT REDUCE GENERATORS OF G yet
                            else
                            end
                            iii=iii+1;
                    end
                    nn;
                    pgroupsize=prod(nn);
                    pgroupsize;









    %ran perm group size program

    if pgroupsize == factorial(n)
        prob=prob+1;
    end
    nnnn=nnnn+1;
end
P_n=prob/10000;
P_n;
probvec(rangeindex)=P_n;
rangeindex=rangeindex+1;
end
probvec









end
```