

TP1 : étude de quelques groupes de permutations

Eléments de corrigé

```
> restart: with(group);
[DerivedS, LCS, NormalClosure, RandElement, SnConjugates, Sylow, areconjugate, center,
centralizer, core, cosets, cosrep, derived, elements, groupmember, grouporder, inter, invperm,
isabelian, isnormal, issubgroup, mulperms, normalizer, orbit, parity, permrep, pres, transgroup
]
```

Question 1

```
> convert([1,3,4,5,2],disjcytc);
[[2,3,4,5]]
> convert([[1,2,3],[4,5]],permlist,5);
convert([[1,2,3],[4,5]],permlist,9);
[2,3,1,5,4]
[2,3,1,5,4,6,7,8,9]
> map(g->convert(g,disjcytc),[[2,3,1,5,4],[2,3,1,5,4,6,7,8,9]]);
[[[1,2,3],[4,5]],[[1,2,3],[4,5]]]
> map(g->convert(g,disjcytc),{[2,3,1,5,4],[2,3,1,5,4,6,7,8,9]});
[[[1,2,3],[4,5]]]
> mulperms([[1,2]],[[1,3]]); mulperms([[1,3]],[[1,2]]);
[[1,2,3]]
[[1,3,2]]
```

La multiplication des permutations sous maple est donc une multiplication à droite : $\text{mulperms}(g1,g2)$ ne donne pas $g1 \log 2$ mais $g2 \log 1$; on définit donc :

```
> multperm:=proc(g1,g2);
return(mulperms(g2,g1));
end:
> multperm([[1,2]],[[1,3]]);
[[1,3,2]]
```

Question 2

```
> combinat[permutate](3);
[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
donne toutes les permutations de la liste [1,2,3], donc tous les éléments de  $S(3)$  en tant que
"permutation lists".
> S3:=permgroupe(3,[[[1,2]],[[1,3]]]);
S3 := permgroupe(3, [[1,2],[1,3]])
> elements(S3);
[[[2,3]],[1],[[1,2]],[[1,3]],[[1,2,3]],[[1,3,2]]]
On convertit les "permutation lists" obtenues par la commande permutate en produit de cycles de
support disjoints, afin de comparer les deux résultats :
> map(i->convert(i,disjcytc),{op(combinat[permutate](3))});
```

```
[[[2,3]],[1],[[1,2]],[[1,3]],[[1,2,3]],[[1,3,2]]]
> elements(S3) minus %;
{}

```

Question 3

```
[ Première définition de  $S(n)$  par générateurs :
> S1:=n->permgroupe(n,{seq([i,i+1],i=1..n-1)});
S1 := n -> permgroupe(n, {seq([i,i+1],i=1..n-1)})
[ On teste pour d'autres valeurs de n :
> seq(elements(S1(j)) minus
map(i->convert(i,disjcytc),{op(combinat[permutate](j))}),j=2..8)
;
{},{},{},{},{},{},{},{ }
[ Noter que le cardinal de  $S(8)$  est déjà :
> 8!;
40320
[ En fait, il est inutile de calculer tous les éléments de  $S1(n)$  ; il suffit de vérifier que son
cardinal est bien  $n!$ , ce qui est plus rapide :
> seq(grouporder(S1(j))/j!,j=2..15);
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
[ Deuxième définition de  $S(n)$  par générateurs (attention au cas  $n=2$  cette fois) :
> S2:=proc(n);
if n=2 then return(permgroupe(2,[[[1,2]]])) else
return(permgroupe(n,[[[1,2]],[[1..n]]])); fi;
end:
> seq(grouporder(S2(j))/j!,j=2..15);
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

```

Question 4

```
> test:=proc(S0) local S,L,g,h,N;
S:=S0; for g in S0 do S:=S union {invperm(g)}; od;
L:=S union {}; N:=S;
N:={seq(seq(multperm(g,h),g=N),h=S)} minus L;
L:=L union N;
print(L,N);
N:={seq(seq(multperm(g,h),g=N),h=S)} minus L;
L:=L union N;
print(L,N);
end:
> test([[1,2]],[[1,2,3]]);
[[[1,2]],[1],[[1,3]],[[1,3,2]],[[1,2,3]],[[2,3]]],[[[1,3]],[[2,3]]]
[[[1,2]],[1],[[1,3]],[[1,3,2]],[[1,2,3]],[[2,3]]],{ }
Après un passage, L contient tous les mots de longueur  $\leq 2$  (en les éléments de  $S$ ) et N les
mots de longueur 2 qui ne peuvent pas s'écrire comme mots de longueur  $< 2$ . L est de cardinal
6, donc  $L=S3$ 
Après un second passage, N est l'ensemble vide.
> test([[1,2]],[[1,2,3,4]]);
[[[1,2]],[1],[[2,3,4]],[[2,4,3]],[[1,4,3,2]],[[1,2,3,4]],[[1,3,4]],[[1,4,3]]]
```

```

[[1, 3], [2, 4]], {[[2, 3, 4], [[2, 4, 3], [[1, 3, 4], [[1, 4, 3]], [[1, 3], [2, 4]]]
{[[1, 2]], [], [[2, 3], [[2, 3, 4], [[2, 4, 3], [[1, 3, 4, 2]], [[1, 4, 3, 2]], [[1, 2, 3, 4],
[[1, 2, 4, 3]], [[1, 3, 4, 4], [[1, 4, 3]], [[1, 3], [2, 4]], [[1, 4, 2, 3]], [[1, 3, 2, 4]], [[1, 4]]},
{[[2, 3]], [[1, 3, 4, 2]], [[1, 2, 4, 3]], [[1, 4, 2, 3]], [[1, 3, 2, 4]], [[1, 4]]}

```

On ne trouve pas $4! = 24$ éléments après deux passages, donc il faut poursuivre plus loin.

```

> elements1:=proc(G) local S,L,g,h,N;
S:=op(2,G); for g in S do S:=S union {invperm(g)}; od;
L:=S union {}; N:=S;
while (nops(N)<>0) do
N:={seq(seq(multperm(g,h),g=N),h=S)} minus L;
L:=L union N;
od;
return(L);
end;
> nops(elements1(S1(4)));

```

24

```

> evalb(nops(elements1(S1(6)))=6!);

```

true

Comparaison des temps de calcul par rapport à la procédure elements de Maple :

```

> t:=time():elements(S1(6)):time()-t;

```

0.032

```

> t:=time():elements1(S1(6)):time()-t;

```

0.232

```

> t:=time():elements(S1(8)):time()-t;

```

6.504

```

> t:=time():elements1(S1(8)):time()-t;

```

37.266

L'algorithme de Maple est bien meilleurs que notre algorithme naïf, qui est impraticable pour n plus grand...

Question 5

```

> parity([[1,2]]; parity([[1,2,3]]);

```

-1

1

```

> parity(S1(3));

```

-1

On obtient 1 si tous les éléments du groupe de permutations sont paires

```

> G:=proc(n);
if n=3 then return(permgroupe(3,[[[1,2,3]]]));
elif type(n,odd) then
return(permgroupe(n,[[[1,2,3]],[[[3..n]]]]));
else return(permgroupe(n,[[[1,2,3]],[[[3..n],[1,2]]]]));
fi;
end;
> seq(parity(G(j)),j=3..15);

```

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

En fait, cette vérification est inutile : comme les générateurs sont pairs, il en sera de même de tous les éléments du groupe engendré.

```

> seq(grouporder(G(n))/n!,n=3..15);

```

$$\frac{1}{2} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'} \frac{1}{2'}$$

On obtient le cardinal du groupe alterné : on conjecture donc qu'il s'agit du groupe alterné.

```

> A:=proc(n);
if n=2 then return(permgroupe(2,[[1]]));
else return(G(n));
fi;
end;

```

Question 6

```

> seq(center(A(n)),n=2..7);

```

```

permgroupe(2, {}), permgroupe(3, {[[1, 2, 3]]}), permgroupe(4, {}), permgroupe(5, {}),

```

```

permgroupe(6, {}), permgroupe(7, {}))

```

```

> elements(A(3));

```

[[], [[1, 2, 3]], [[1, 3, 2]]]

```

> isabelian(A(3));

```

true

Pour $n \geq 4$, le centre de $A(n)$ est trivial ; c'est facile à démontrer...

Question 7

```

> typ:=proc(n,g) local g1,N,res,i,N1;
if not type(g,disjycyc(n)) then g1:=convert(g,disjycyc) else
g1:=g; fi;
N:=nops(g1);
if N=0 then return([1$N])
else res:=[seq(nops(op(i,g1)),i=1..N)];
N1:=sum(res[i],i=1..nops(res));
return([1$(n-N1),op(sort(res))]);
fi;
end;

```

```

> typ(6,[[1,2,3]]); typ(6,[[1,2,3],[4,5]]);

```

[1, 1, 1, 3]

[1, 2, 3]

```

> ord:=proc(n,g)
return(ilcm(op(typ(n,g))));
end;

```

Voici la liste des ordres des éléments de S_4 :

```

> map(g->ord(4,g),elements(S1(4)));

```

{1, 2, 3, 4}

C'est un ensemble de diviseurs de 24.

```

> sort(map(g->ord(4,g),[op(elements(S1(4)))]));

```

[1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4]

Il y a en fait 9 éléments d'ordre 2, 8 d'ordre 3 et 6 d'ordre 4

Question 8

Nuérôtant les sommets à partir du sommet en haut à gauche, dans le sens des aiguilles d'une

```
montre, on a :
> R1:=[[1,2,3,4]]: R2:=multperm(R1,R1): R3:=multperm(R2,R1):
  H:=[[1,4],[2,3]]: V:=[[1,2],[4,3]]: Delta1:=[[1,3]]:
  Delta2:=[[2,4]]:
> D4:=permgroupe(4,{R1,V,H,Delta1,Delta2}):
> elements(D4);
{
  [[1,2],[3,4]], [[1,4],[2,3]], [], [[1,3]], [[2,4]], [[1,3],[2,4]], [[1,4,3,2]], [[1,2,3,4]]
}
Ce sont les 8 éléments précédents : ces éléments engendrent un groupe d'ordre 8, donc
constituent bien un groupe.
```

Question 9

```
> map(G->groupeorder(G), {permgroupe(4, {R1,H}), permgroupe(4, {R1,V}),
  permgroupe(4, {R1,Delta1}), permgroupe(4, {R1,Delta2})});
{8}
Ces couples d'éléments engendrent tous un groupe d'ordre 8, donc D4 tout entier.
> for g in {H,V,Delta1,Delta2} do
  evalb(multperm(g,R1)=multperm(invperm(R1),g)); od;
true
true
true
true
Dans chaque cas, on a D4=<a,b>, avec a d'ordre 4, b d'ordre 2 et ba=a^3b, ces relations
suffisant à écrire la table du groupe.
On avait d'autres possibilités (R3 est également d'ordre 4) :
> map(G->groupeorder(G), {permgroupe(4, {R3,H}), permgroupe(4, {R3,V}),
  permgroupe(4, {R3,Delta1}), permgroupe(4, {R3,Delta2})});
{8}
Recherchons toutes les possibilités :
> L:=elements(D4): res:={}: for a in L do for b in L do if
  groupeorder(permgroupe(4,{a,b}))=8 then res:=res union {{a,b}};
  fi; od; od; res;
{{{[1,4],[2,3]],[[1,3]]}, {[[1,4],[2,3]],[[2,4]]}, {[[1,2],[3,4]],[[1,3]]},
  {[[1,2],[3,4]],[[2,4]]}, {[[1,2],[3,4]],[[1,4,3,2]]}, {[[1,2],[3,4]],[[1,2,3,4]]},
  {[[1,3]],[[1,2,3,4]]}, {[[2,4]],[[1,2,3,4]]}, {[[1,4],[2,3]],[[1,2,3,4]]},
  {[[1,4],[2,3]],[[1,4,3,2]]}, {[[2,4]],[[1,4,3,2]]}, {[[1,3]],[[1,4,3,2]]}}
Il y a donc 12 couples de générateurs engendrant D4. On peut engendrer D4 avec deux
éléments d'ordre 2 :
> a:=[[1,2],[3,4]]: b:=[[1,3]]: c:=multperm(a,b):
  d:=multperm(b,a): multperm(c,a): multperm(d,b):
  multperm(c,c):
c:=[[1,4,3,2]]
d:=[[1,2,3,4]]
  [[2,4]]
  [[1,4],[2,3]]
  [[1,3],[2,4]]
```

Ainsi $D4 = \langle a, b \rangle$, où a et b sont d'ordre 2 et $(ab)^2 = (ba)^2$, ces relations définissant complètement le groupe.

Question 10

```
> with(LinearAlgebra):
> a:=Matrix([[0,1],[-1,0]]): b:=Matrix([[0,I],[I,0]]):
a := [ 0  1 ]
      [-1 0 ]
b := [ 0  I ]
      [ I  0 ]
> a^2; b^2; c:=a.b;
c := [ -1  0 ]
      [ 0  -1 ]
      [-1  0 ]
      [ 0  -1 ]
      [ I  0 ]
      [ 0  -I ]
```

Donc $H = \{ \pm I, \pm a, \pm b, \pm c \}$. En choisissant de numéroter dans l'ordre $I, a, b, c, -I, -a, -b, -c$, on obtient comme permutations correspondant à a et b respectivement :

```
> g:=convert([2,5,4,7,6,1,8,3],disjcy);
  h:=convert([3,8,5,2,7,4,1,6],disjcy);
g := [[1,2,5,6],[3,4,7,8]]
h := [[1,3,5,7],[2,8,6,4]]
```

Comme $H = \langle a, b \rangle$, on définit :

```
> H:=permgroupe(8,{g,h}):
> groupeorder(H);
8
```

On trouve 8 éléments, donc H est bien un groupe.

```
> multperm(h,g): multperm(g,invperm(h));
[[1,8,5,4],[2,7,6,3]]
[[1,8,5,4],[2,7,6,3]]
```

Donc $hg = gh^3$, relation que l'on aurait pu vérifier également matriciellement :

```
> b^(-1).a;
[ I  0 ]
[ 0 -I ]
```

Question 11

```
> a:=[[1,2,3,4],[5,6,7,8]]: b:=[[1,5,3,7],[2,8,4,6]]:
> groupeorder(permgroupe(8,{a,b}));
8
> multperm(a,a): multperm(b,b): multperm(a,b):
  multperm(invperm(b),a);
```

```
[[1, 3], [2, 4], [5, 7], [6, 8]]
```

```
[[1, 3], [2, 4], [5, 7], [6, 8]]
```

```
[[1, 6, 3, 8], [2, 5, 4, 7]]
```

```
[[1, 6, 3, 8], [2, 5, 4, 7]]
```

[Les relations caractérisant H sont vérifiées, donc $\langle a, b \rangle$ est isomorphe à H.

[Connaissant la classification des groupes de cardinal 8, on aurait pu conclure comme suit :

```
> isabelian(permgroupe(8, {a,b}));
```

```
false
```

[En effet, a et b ne commutent pas. Donc il s'agit de D4 ou H

```
> map(g->ord(8,g), [op(elements(permgroupe(8, {a,b})))]);
```

```
[4, 4, 4, 4, 4, 1, 2, 4]
```

[On dénombre 6 éléments d'ordre 4 : il s'agit donc de H.

Question 12

[La procédure suivante teste si deux éléments a et b d'ordre 4 engendrent un sous-groupe isomorphe à H :

```
> test:=proc(a,b):  
  if multperm(a,a)=multperm(b,b) and  
  multperm(a,b)=multperm(invperm(b),a) then return(true) else  
  return(false); fi;  
end:  
> test(a,b);
```

```
true
```

[On détermine la liste des éléments d'ordre 4 dans S7 :

```
> G:=elements(S1(7)): L:={}:  
  for g in G do if ord(7,g)=4 then L:=L union {g};fi; od:  
> nops(L);
```

```
840
```

```
> for a in L do for b in L do if test(a,b) then print(a,b);  
  break; fi; od;od;
```

[Il n'y en n'a pas !