Pari-Befehle zum Rechnen mit Permutationen

Wir fassen Permutationen in $\mathfrak{S}_n$ als Listen von n Elementen auf, wobei alle Listenelemente verschieden sind und zwischen 1 und n liegen.

Das Produkt von zwei Permutationen sieht dann in Pari so aus:

```
perm_mult(p,q)=local(n=length(p),r=vector(n)); \
      if(n!=length(q),return("p und q besitzen nicht die gleiche Länge"));\
\
      for(i=1,n,r[i]=p[q[i]]);\
\
return(r)
```

Die folgende Funktion gibt die Identität in $\mathfrak{S}_n$ zurück:

```
perm_id(n)=return(vector(n,i,i))
```

Die Inversenberechnung sieht so aus:

```
perm_inverse(p)=local(n=length(p),q=vector(n));for(i=1,n,q[p[i]]=i);return(q)
```

Für relativ kleine Potenzen einer Permutation eignet sich

```
perm_pow(p,n)=    if(n==0,return(vector(length(p),i,i)));\
                  if(n>0,return(perm_mult(p,perm_pow(p,n-1))));\
                  if(n<0,return(perm_pow(perm_inverse(p),-n)))
```

Relativ wichtig scheint mir auch die Herstellung einer zufälligen Permutation in $\mathfrak{S}_n$:

```
perm_random(n)=local(p=vector(n,i,i),tmp,j);\
            for(i=1,n,tmp=p[i];p[i]=p[j=random(n)+1];p[j]=tmp);return(p)
```

Dieselbe Funktionalität hat

```
perm_random_recursive(n)=local(i=random(n)+1,p);\
            if(n==1,return([1]),\
                  p=concat(perm_random_recursive(n-1),[0]));\
                  p[n]=p[i];p[i]=n;\
return(p)
```

Ist irgendwie hübscher als `perm_random`!

Mit diesen Werkzeugen sollten viele Fragen zu Permutationen leicht zu beantworten sein.
Beachten Sie, daß in der Definition von `perm_pow` die Funktion `perm_mult` benutzt wird; letztere muß deshalb vorher definiert sein.

Beispiel:

```
(15:34) gp > p=perm_random(10)
%260 = [4, 8, 2, 6, 5, 1, 3, 10, 9, 7]
(15:35) gp > q=p
%261 = [4, 8, 2, 6, 5, 1, 3, 10, 9, 7]
(15:35) gp > q=perm_mult(p,q)
%262 = [6, 10, 8, 1, 5, 4, 2, 7, 9, 3]
(15:35) gp > q=perm_mult(p,q)
%263 = [1, 7, 10, 4, 5, 6, 8, 3, 9, 2]
(15:35) gp > q=perm_mult(p,q)
%264 = [4, 3, 7, 6, 5, 1, 10, 2, 9, 8]
(15:35) gp > q=perm_mult(p,q)
%265 = [6, 2, 3, 1, 5, 4, 7, 8, 9, 10]
(15:35) gp > q=perm_mult(p,q)
%266 = [1, 8, 2, 4, 5, 6, 3, 10, 9, 7]
(15:36) gp > q=perm_mult(p,q)
%267 = [4, 10, 8, 6, 5, 1, 2, 7, 9, 3]
(15:36) gp > q=perm_mult(p,q)
%268 = [6, 7, 10, 1, 5, 4, 8, 3, 9, 2]
(15:36) gp > q=perm_mult(p,q)
%269 = [1, 3, 7, 4, 5, 6, 10, 2, 9, 8]
(15:36) gp > q=perm_mult(p,q)
%270 = [4, 2, 3, 6, 5, 1, 7, 8, 9, 10]
(15:36) gp > q=perm_mult(p,q)
%271 = [6, 8, 2, 1, 5, 4, 3, 10, 9, 7]
(15:36) gp > q=perm_mult(p,q)
%272 = [1, 10, 8, 4, 5, 6, 2, 7, 9, 3]
(15:36) gp > q=perm_mult(p,q)
%273 = [4, 7, 10, 6, 5, 1, 8, 3, 9, 2]
(15:36) gp > q=perm_mult(p,q)
%274 = [6, 3, 7, 1, 5, 4, 10, 2, 9, 8]
(15:36) gp > q=perm_mult(p,q)
%275 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

(15:39) gp > perm_pow(p,15)
%276 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Man sieht, daß die Permutation $p=(4,8,2,6,5,1,3,10,9,7)\in\mathfrak{S}_{10}$ die Ordnung 15 besitzt!
Wer möchte dies per Hand so nachrechnen?

```
(15:41) gp > perm_inverse(p)
%277 = [6, 3, 7, 1, 5, 4, 10, 2, 9, 8]
```

Dieses Ergebnis für p^{-1} ist vernünftigerweise dasselbe wie `%274`, also p^{14}.

Zykelschreibweise für Permutationen

Mit etwas Überlegung kann übrigens man doch leicht sehen, daß die obige Permutation $p\in\mathfrak{S}_{10}$ die Ordnung 15 besitzt:

Man bilde die Folge 1, $p(1)=4$, $p(4)=6$, $p(6)=1$. Wegen der Bijektivität von p muß sich dabei ein „Zykel" ausbilden, in diesem Fall [1,4,6]. Schauen wir uns den Zykel an, der durch p von 2 ausgehend erzeugt wird: [2,8,10,7,3]. Es fehlt nur noch die 5, die auf sich selbst abgebildet wird, also einen einelementigen Zykel [5] bildet. Jeden dieser Zykel können wir mit einer Permutation identifizieren, indem wir die Elemente, die im Zykel nicht vorkommen, auf sich selbst abbilden:

Damit entsprechen diese Zykel im obigen Beispiel den folgenden Permutationen
s=[1,4,6]=(4,2,3,6,5,1,7,8,9,10)
t=[2,8,10,7,3]=(1,8,3,4,5,6,3,10,9,7)
e=[5]=(1,2,3,4,5,6,7,8,9,10) , dies ist die Identität.

und die Ausgangspermutation selbst ist Produkt der Zykel, also $p=s\circ t=t\circ s$.

Wichtig ist, daß Permutationen, die verschiedenen Zykeln entsprechen, kommutieren, daß heißt bei ihrer Multiplikation kommt es auf die Reihenfolge nicht an. Dies überlegt man sich leicht, kann es aber im obigen Beispiel auch leicht verifizieren.

Man sieht auch sofort, daß die einem Zykel der Länge *k* entsprechende Permutation die Ordnung *k* besitzt. Damit ist auch klar, daß $p=s\circ t$ die Ordnung kgV(ord s , ord *t*) besitzt, in unserem Fall also 15. Man rechnet ja auch leicht nach, daß $p^{15}=(s\circ t)^{15}=s^{15}\circ t^{15}=(s^3)^5\circ(t^5)^3=e^3\circ e^5=e$, was für sich bereits bedeutet, daß die Ordnung von *p* ein Teiler von 15 ist. Das zweite Gleichheitszeichen ist wegen der Kommutativität von *s* und *t* gerechtfertigt: es ist ja

$$(s\circ t)^{15}=s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t\circ s\circ t=$$
$$s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ s\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t\circ t=s^{15}\circ t^{15}\ .$$

Im Allgemeinen kommutieren verschiedene Permutationen natürlich nicht!