RCC-Dets.mw

```
> #with(PolynomialTools);with(DiscreteTransforms);

> read `/home/ph/maple/RC.txt`:#read
  "/home/ph/maple/cartan.txt";unprotect(gamma);
```

Warning, the names GramSchmidt and fibonacci have been rebound

Warning, the name GramSchmidt has been rebound

Warning, the name fibonacci has been rebound

Warning, `rvec` is implicitly declared local to procedure `goCycles`

```
> ############ START HERE ######################

> Delta:=evalm((R-B)/2):A:=evalm((R+B)/2):
  `R`=matrans(R);`B`=matrans(B);n;

> unassign('e','w','t','k'):wpi:=clearDenoms(evalm(1/w[1]*linsolve(J-transpose(A

> pi:=evalm(wpi/W):omega:=diag(seq(pi[k],k=1..n)):"PI-CHECK",iszero(symm

> Omega:=stackmatrix(seq(pi,k=1..n)):`pi`=evalm(pi);

> nsd:=nullspace(Delta):
        print(`NullSpace  of Delta `,nullity(Delta));
        print(seq(readVec(nsd[k]),k=1..nullity(Delta)));
              nsk:=nullspace(Delta):
        print(`Nullspace of A`,nullity(A));
        NSA:=nullspace(A):
        if(det(A)=0) then
              for ix to nops(NSA) do
        nx:=NSA[ix];
        P:={};Q:={};
              for aa to n do
```

```
                if(nx[aa]=1) then P:={op(P),aa} fi;
                if(nx[aa]=-1) then Q:={op(Q),aa} fi;
            od;
            print([P,Q]);
        od;
        else
            print(` det(A) = `);
            fracfm(det(A))
        fi;
```

```
> if(nullity(A)=nullity(Delta)) then "CC" else "NOT CC" fi;
```

$$R = [3, 3, 1, 1, 7, 7, 5, 5]$$

$$B = [6, 8, 8, 6, 2, 4, 4, 2]$$

$$8$$

$$wpi := [1, 1, 1, 1, 1, 1, 1, 1]$$

$$W := 8$$

"PI-CHECK", *true*

$$\pi = \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right]$$

NullSpace of Delta, 2

{2, 4, 5, 7}, {1, 3, 6, 8}

Nullspace of A, 2

[{6, 8}, {1, 3}]

[{5, 7}, {2, 4}]

"CC"

```
> unassign('f'):f:=vector(n);ff:=diag(seq(f[k],k=1..n)):PA:=multiply(omega,A)
```

$f := array(1..8, [])$

true

CYCLE DECOMPOSITION HERE

```
> ccls:=goCycles(PA):

> AP:=evalm(add(ccls[k][1]/nops(ccls[k][2])*cyc(ccls[k][2]),k=1..nops(ccls)))
  A",iszero(AP-omega&*A);

> cvx:=seq(ccls[k][1],k=1..nops(ccls)):

> vcx:=clearDenoms(evalm(Vector([cvx])));if(`+`(cvx)=1) then
  print("convexchex") else print("ERROR!")
  fi;WCYC:=add(vcx[k],k=1..nops(ccls));

> print(`
  `);print("=====================================
  `);

> "R=",matrans(R),"B=",matrans(B);print("Values,Cycles",ccls);print("Friedman
  weights",wpi,cat(` W = `,add(wpi[k],k=1..n),` with WCYC = `,WCYC));
```

"finding cycles"

"found cycle of length", 2

"found cycle of length", 3

"found cycle of length", 3

"found cycle of length", 3

"found cycle of length", 3

"found cycle of length", 2

"Reconstituting A", *true*

$vcx := [3, 3, 2, 2, 3, 3]$

"convexchex"

$WCYC := 16$

"==

"R=", [3, 3, 1, 1, 7, 7, 5, 5], "B=", [6, 8, 8, 6, 2, 4, 4, 2]

$$\text{"Values,Cycles"}, \left\{ \left[\frac{3}{16}, [1, 6, 4] \right], \left[\frac{3}{16}, [2, 3, 8] \right], \left[\frac{1}{8}, [5, 7] \right], \left[\frac{1}{8}, [1, 3] \right], \left[\frac{3}{16}, [4, 6, 7] \right], \left[\frac{3}{16}, [2, 8, 5] \right] \right\}$$

"Friedman weights", [1, 1, 1, 1, 1, 1, 1, 1], $W = 8$ *with* $WCYC = 16$

SAVING R, B, and CYCLE DATA

```
> outfile:=cat("/home/ph/maple/CYCLES",matrans(R)[],".m");save(R,B,ccls,out
```

outfile := "/home/ph/maple/CYCLES33117755.m"

READING CYCLE DATA. FIRST RENAME FILE TO CYCLESn.m WHERE n=# of VERTICES

```
> #outfile:=cat("/home/ph/maple/CYCLES",n,".m");read (outfile);ccls;

> ########################################

> seq(ccls[k][1],k=1..nops(ccls)):if(1<>`+`(%)) then "Error. Recheck
  calculations."
  fi;Digits:=2:print(seq([k,evalf(ccls[k][1])],k=1..nops(ccls)));Digits:=10:
```

[1, 0.12], [2, 0.19], [3, 0.19], [4, 0.19], [5, 0.19], [6, 0.12]

```
> for i to nops(ccls) do
  prob[i]:=add(pi[ccls[i][2][j]],j=1..nops(ccls[i][2]));od:mx:=max(seq(prob[i
```

```
j to nops(ccls) do if(prob[j]=mx) then MX:=[op(MX),j];print("max prob cycle'
```

```
> mx:=min(seq(prob[i],i=1..nops(ccls))):"min",mx;for j to nops(ccls) do
  if(prob[j]=mx) then print("min prob cycle",j,ccls[j]) fi
  ;od;Digits:=2:print(seq([k,evalf(prob[k])],k=1..nops(ccls)));Digits:=10:
```

$$\text{"max"}, \frac{3}{8}$$

$$\text{"max prob cycle"}, 2, \left[\frac{3}{16}, [2, 8, 5]\right]$$

$$\text{"max prob cycle"}, 3, \left[\frac{3}{16}, [2, 3, 8]\right]$$

$$\text{"max prob cycle"}, 4, \left[\frac{3}{16}, [1, 6, 4]\right]$$

$$\text{"max prob cycle"}, 5, \left[\frac{3}{16}, [4, 6, 7]\right]$$

$$\text{"min"}, \frac{1}{4}$$

$$\text{"min prob cycle"}, 1, \left[\frac{1}{8}, [1, 3]\right]$$

$$\text{"min prob cycle"}, 6, \left[\frac{1}{8}, [5, 7]\right]$$

$$[1, 0.25], [2, 0.38], [3, 0.38], [4, 0.38], [5, 0.38], [6, 0.25]$$

Now color the maximum likelihood cycle Red.

```
> j:=MX[1]:print(ccls[j]):CYMAX:=evalm(cyc(ccls[j][2])):rr:=readrows(CYMA
  k to n do if(rr[k]<>[]) then
  Delta:=rowreplace(Delta,k,evalm(2*Delta[k,op(rr[k])]*row(Delta,k))) fi; od:
```

$$\left[\frac{3}{16}, [2, 8, 5]\right]$$

```
> j:=1:print(ccls[j]):CY1:=evalm(cyc(ccls[j][2])):rr:=readrows(CY1):for k to
  n do if(rr[k]<>[]) then
```

```
Delta:=rowreplace(Delta,k,evalm(2*Delta[k,op(rr[k])]*row(Delta,k))) fi;
od:
```

```
> #iszero(multiply(CYMAX,CY1));
```

$$\left[\frac{1}{8}, [1, 3]\right]$$

```
> #evalm(PA):mx:=1:
        #for i to nd do
                #for j to nd do
                        #if(PA[i,j]>0 and PA[i,j]<mx) then
                        #       mx:=PA[i,j];
                        #       ix:=i;
                        ##       jx:=j;
                        #fi
                #od:      od:
```

```
> #mx;
```

```
> #ccx:=[11,1,9,13]:for k to nops(ccx) do k1:=k+1: if(k=nops(ccx))then
  k1:=1 fi; vl:=Delta[ccx[k],ccx[k1]]:if(vl<>1) then
  Delta:=rowreplace(Delta,ccx[k],evalm(2*vl*row(Delta,ccx[k]))) fi; od:
```

```
> R:=evalm(1/2*(A+Delta)):B:=evalm(1/2*(A-Delta)):
```

RECOLORING STARTS HERE

```
> #colrs:={{1,2,25,30,4,32,6, 11, 13, 17, 24, 28, 34,36}};
```

```
>
```

```
> colrs:={{2,3,7}};
```

```
> unassign('COLR','e','f'):print("wpi",wpi);print("pi",pi);
```

```
> ee:=vector(n,1):KOL:=1:COLR:=colrs[KOL];for i to nops(COLR) do
  ee[convert(COLR[i],decimal,hex)]:=-1
```

```
od:print(map(convert,COLR,decimal,hex));

> phi:=diag(seq(ee[q],q=1..n)):Detta:=evalm(phi&*Delta):R:=evalm(A+Detta

> unassign('t'):print();

> for t in [1,-1] do print();print();if(t=1) then print("R") else print("B")
  fi;X:=evalm(A+t*Detta):print(matrans(X));print(map(convert,matrans(X),dec

> if(tailchk(X)) then print("tailcheck")
   fi;rk:=rank(X):rca:=op(readcycles(abel(X)));rcc:=`union`(rca);print();print(

> KK:=pipow(pi,X);

> print();print(rank(KK),"vs",rk);print();print();

> if(rank(KK)=rk) then

> YZ:=evalm(X^rk):rx:=rank(YZ);KXK:=submatrix(KK,rk-rx+1..rk,[op(rcc)]):]

> print("with cycle rank",rx) ;print("DET",det(KXK));

> vv:=Vector(row(KXK,1));

> FT:=FourierTransform(vv);print("Fourier Transform"):for i to rowdim(KXK)
  do print(i,FT[i]) od;

> fi;

> for j to rank(X) do
  Y:=evalm(X^j);KK:=pipow(pi,Y):;print(readrows(Y),rank(KK),"vs",rank(Y));

> if(iszero(Y-Y^2)) then print(j,"th power is idempotent, checking next
  one");break fi;od;

> Y:=evalm(X&*Y):KK:=pipow(pi,Y):print(KK);print(readrows(Y),rank(KK),"vs'
```

> od:

> dot(YY[1],YY[-1]):

$colrs := \{\{2, 3, 7\}\}$

"wpi", [1, 1, 1, 1, 1, 1, 1, 1]

$\text{"pi"}, \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right]$

$COLR := \{2, 3, 7\}$

{2, 3, 7}

"R"

[*3*, *8*, *8*, *1*, *7*, *7*, *4*, *5*]

[3, 8, 8, 1, 7, 7, 4, 5]

"Cycles", {1, 3, 4, 5, 7, 8}

6, "vs", 6

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$$

"with cycle rank", 6

"DET", $\frac{-1}{16384}$

"Fourier Transform"

1, 0.408248290463863018 + 0. *I*

2, 0. + 0.0883883476483184327 *I*

3, −0.0510310363079828772 + 0. *I*

4, 0. + 0. *I*

5, −0.0510310363079828772 + 0. *I*

6, 0. − 0.0883883476483184327 *I*

[[3], [8], [8], [1], [7], [7], [4], [5]], 6, "vs", 6

[[8], [5], [5], [3], [4], [4], [1], [7]], 3, "vs", 6

[[5], [7], [7], [8], [1], [1], [3], [4]], 2, "vs", 6

[[7], [4], [4], [5], [3], [3], [8], [1]], 3, "vs", 6

[[4], [1], [1], [7], [8], [8], [5], [3]], 6, "vs", 6

[[1], [3], [3], [4], [5], [5], [7], [8]], 1, "vs", 6

6, "th power is idempotent, checking next one"

$$\begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{8} & 0 & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & 0 & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$$

[[3], [8], [8], [1], [7], [7], [4], [5]], 6, "vs", 6

"B"

$[6, 3, 1, 6, 2, 4, 5, 2]$

[6, 3, 1, 6, 2, 4, 5, 2]

"Cycles", {4, 6}

5, "vs", 6

[[6], [3], [1], [6], [2], [4], [5], [2]], 5, "vs", 6

[[4], [1], [6], [4], [3], [6], [2], [3]], 3, "vs", 5

[[6], [6], [4], [6], [1], [4], [3], [1]], 2, "vs", 4

[[4], [4], [6], [4], [6], [6], [1], [6]], 2, "vs", 3

[[6], [6], [4], [6], [4], [4], [6], [4]], 1, "vs", 2

[[4], [4], [6], [4], [6], [6], [4], [6]], 1, "vs", 2

6, "th power is idempotent, checking next one"

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

[[6], [6], [4], [6], [4], [4], [6], [4]], 1, "vs", 2

```
> q:=[1,2,6,4,5,3];Q:=matrix(rx,rx,(i,j)->if(j=q[i]) then 1 else 0 fi);
```

$$q := [1, 2, 6, 4, 5, 3]$$

$$Q := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

```
> rx;jj:=IdentityMatrix(rx):pp:=evalm(1/x[1]*linsolve(jj-transpose(KXK),vector
```

6

$$PK := \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$$

```
> KPK:=multiply(PK,Q);
```

```
> vv:=Vector(row(KPK,1));
```

```
> FT:=FourierTransform(vv):print("Fourier Transform"):for i to rowdim(KXK)
  do print(i,FT[i]) od;
```

```
>
```

$$KPK := \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix}$$

$$vv := \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8} \right]$$

"Fourier Transform"

$1,\ 0.408248290463863018 + 0.\,I$

$2,\ -0.0510310363079828772 + 0.\,I$

$3,\ -0.0510310363079828772 + 0.\,I$

$4,\ 0.102062072615965754 + 0.\,I$

$5,\ -0.0510310363079828772 + 0.\,I$

$6,\ -0.0510310363079828772 + 0.\,I$

$\{\}$

```
> dot(R,B);
```

$$0$$

SAVING COLORING

```
> #outfile:=cat("/home/ph/maple/RANKED",matrans(R)[],".m");save(R,B,outfi

> "CHARS",`R`,charpoly(R,s),`B`,charpoly(B,s),"MINS",`R`,minpoly(R,s),`B`,m

> AR:=abel(R):AB:=abel(B):print("AR",AR,"AB",AB);"u's",multiply(AR,u),multip

> print();print("nullity of A",nullity(A),"nullity of Delta",nullity(Delta));

> if(N2<>binomial(n,2)) then
  N2:=binomial(n,2):u:=vector(n,1):uu:=vector(N2,1):liecliff(n):UJ:=matrix(r

> R2:=sympow(R,2):B2:=sympow(B,2):A2:=evalm(1/2*(R2+B2)):

> dt:=det(J2-A2):if(dt<>0) then "SYNC'D",dt else "NOT SYNC'D" fi;print();
```

$$\text{"CHARS"}, R, s^8 - s^2, B, s^8 - s^6, \text{"MINS"}, R, -s + s^7, B, -s^5 + s^7$$

$$"AR", \begin{bmatrix} \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} \end{bmatrix}, "AB", \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

"u's", [1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1]

$$"pi's", \left[\frac{4}{3}, 0, \frac{4}{3}, \frac{4}{3}, \frac{4}{3}, 0, \frac{4}{3}, \frac{4}{3}\right], [0, 0, 0, 4, 0, 4, 0, 0]$$

"nullity of A", 2, "nullity of Delta", 2

$$"SYNC'D", \frac{2665}{65536}$$

```
> "u2",clearDenoms(evalm(1/x[1]*linsolve(J2-A2,vector(N2,0),'r',x)));print();#
> unassign('x'):pr2:=clearDenoms(evalm(1/x[1]*linsolve(J2-transpose(R2),vect
> if (nops(nullspace(J2-R2))>1) then
> ar:=abel(R2):
> ur2:=clearDenoms(multiply(ar,uu));pr2:=clearDenoms(multiply(uu,ar));
> fi:
```

```
> print("ur2",evalm(ur2));print("pr2",pr2);print();

> unassign('x'):pb2:=clearDenoms(evalm(1/x[1]*linsolve(J2-transpose(B2),vect

> if (nops(nullspace(J2-B2))>1) then

> ab:=abel(B2):

> ub2:=clearDenoms(multiply(ab,uu));pb2:=clearDenoms(multiply(uu,ab));

> fi:

> print("ub2",evalm(ub2));print("pb2",pb2);
```

```
"u2", [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

"ur2", [1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1]

"pr2", [0, 10, 10, 10, 0, 11, 11, 0, 0, 0, 0, 0, 0, 11, 11, 0, 10, 10, 11, 0, 10, 10, 0, 10, 10, 0, 0, 11]

"ub2", [0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1]

"pb2", [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
> m:=max(seq(ur2[k],k=1..N2)):NXR:=matvec(evalm(ur2/m)):NR:=(evalm(U

> r1:=trace(UJ&*NXR):M:=matvec(pr2):r2:=trace(M&*UJ):MR:=evalm(r1/r2

> gR:=goRange(NR):partR:=[]:for i to rank(R) do
  partR:=[op(partR),readVec(map(diff,gR,y[i]))]
  od:print("R-partitions",partR);

> gB:=goRange(NB):partB:=[]:for i to rank(B) do
  partB:=[op(partB),readVec(map(diff,gB,y[i]))]
  od:print("B-partitions",partB);

> "NMR",multiply(NXR,MR),"NMB",multiply(NXB,MB);iszero(cmm(AR,%[2])),is
```

```
probabilities";multiply(pi,gR),"        ",multiply(pi,gB);

> print();print(` ==========================
`);print();print();

> #"R-OmegaNMR",multiply(u,AR,MR),"B-OmegaNMB",multiply(u,AB,MB);print

> #multiply(AR,NXR),multiply(u,AR,NXR),multiply(AB,NXB),multiply(u,AB,NXI

> iszero(symult(R,NXR)-NXR),iszero(symult(B,NXB)-NXB);iszero(symmult(R,MI
```

"R-partitions", [{1}, {2, 3}, {4}, {5, 6}, {7}, {8}]

"B-partitions", [{3, 5, 6, 8}, {1, 2, 4, 7}, {}, {}, {}, {}]

$$\text{"NMR"}, \begin{bmatrix} \frac{26}{3} & 0 & 7 & 7 & 7 & 0 & \frac{41}{6} & \frac{41}{6} \\ 7 & 0 & \frac{26}{3} & \frac{41}{6} & \frac{41}{6} & 0 & 7 & 7 \\ 7 & 0 & \frac{26}{3} & \frac{41}{6} & \frac{41}{6} & 0 & 7 & 7 \\ 7 & 0 & \frac{41}{6} & \frac{26}{3} & \frac{41}{6} & 0 & 7 & 7 \\ 7 & 0 & \frac{41}{6} & \frac{41}{6} & \frac{26}{3} & 0 & 7 & 7 \\ 7 & 0 & \frac{41}{6} & \frac{41}{6} & \frac{26}{3} & 0 & 7 & 7 \\ \frac{41}{6} & 0 & 7 & 7 & 7 & 0 & \frac{26}{3} & \frac{41}{6} \\ \frac{41}{6} & 0 & 7 & 7 & 7 & 0 & \frac{41}{6} & \frac{26}{3} \end{bmatrix}, \text{"NMB"}, \begin{bmatrix} 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16 & 0 \end{bmatrix}$$

true, *true*

"cycle probabilities"

$$\frac{1}{8} y_1 + \frac{1}{4} y_2 + \frac{1}{8} y_3 + \frac{1}{4} y_4 + \frac{1}{8} y_5 + \frac{1}{8} y_6, \text{"}\quad\text{"}, \frac{1}{2} y_2 + \frac{1}{2} y_1$$

==========================

true, true

true, true

```
> print(gR);
```

$[y_1, y_2, y_2, y_3, y_4, y_4, y_5, y_6]$

```
> #kr:=4:kb:=2:RK:=sympow(R,kr):BK:=sympow(B,kb):ar:=abel(RK):ab:=ab

> #urk:=clearDenoms(multiply(ar,uur));prk:=clearDenoms(multiply(uur,ar));u

> #print(seq([k,choose(n,kr)[k]],k=1..binomial(n,kr)));print(seq([k,choose(n,k

> #rank(R);rank(submatrix(pipow(pi,R),1..rank(R),[7,9,10,19,23,32]));

> #unassign('CL'):for i to nops(ccls) do
  CL[i]:=evalm(1/nops(ccls[i][2])*inverse(omega)&*cyc(ccls[i][2]));ccls[i][1],
  #od;AC:=evalm(add(ccls[i][1]*CL[i],i=1..nops(ccls))):iszero(AC-A);
```