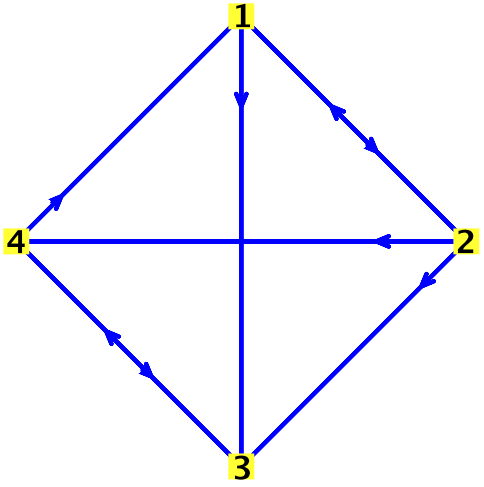
Random walks in directed graphs.

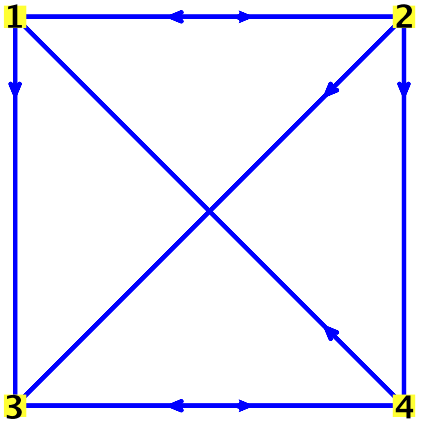
```
> with(GraphTheory):
> G := Graph( directed, [1,2,3,4], { [1,2], [2,4], [4,3], [3,4], [4,1],
  [1,3], [2,1], [2,3] } );
```

G := Graph 1: a directed unweighted graph with 4 vertices and 8 arc(s)

```
> DrawGraph(G);
```




```
> SetVertexPositions(G,[[0,1],[1,1],[0,0],[1,0]]);
> DrawGraph(G);
```




```
> IsStronglyConnected(G);
```

true

```
> AllPairsDistance(G);
```

$$\begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}$$

```
> Neighbors(G);
```

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

```
> Arrivals(G);
```

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

```
> Departures(G);
```

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

```
> R[1] := rand(1..1): # for P3
  R[2] := rand(1..2): # for P1 and P4
  R[3] := rand(1..3): # for P2
```

```
> x := 1; # starting page
  N := 10; # the number of steps
  W := Array(1..N): # the random walk
  DG := Departures(G);
  for k to N do
     Nx := DG[x];
     n := nops(Nx); # number of arcs from vertex x
      r := R[n]();
     x := Nx[r]; # new page
      W[k] := x;
  od:
```

$$x := 1$$

$$N := 10$$

$$DG := [[2, 3], [1, 3, 4], [4], [1, 3]]$$

```
> W;
```

$$\begin{bmatrix} 2 & 4 & 1 & 3 & 4 & 1 & 3 & 4 & 3 & 4 \end{bmatrix}$$

```
> F := Vector(4):
  for k to N do x := W[k]; F[x] := F[x]+1; od:
```

```
> V := F/N*1.0;
```

$$V := \begin{bmatrix} 0.200000000000000 \\ 0.100000000000000 \\ 0.300000000000000 \\ 0.400000000000000 \end{bmatrix}$$

Arrays can have 1, 2, 3, or more dimensions and indexing may start at any value

```
> A := Array(0..2,2..3);
```

$$A := Array(0..2, 2..3, \{\}, datatype = anything, storage = rectangular, order = Fortran\_order)$$

```
> ArrayDims(A);
```

$$0..2, 2..3$$

By default values are initialized to 0.

```
> A[0,2];
```

$$0$$

```
> A[0,2] := 2;
```

$$A_{0,\,2} := 2$$

```
> A[0,2];
```

$$2$$

```
> L := convert(A,listlist);
```

$$L := [[2, 0], [0, 0], [0, 0]]$$

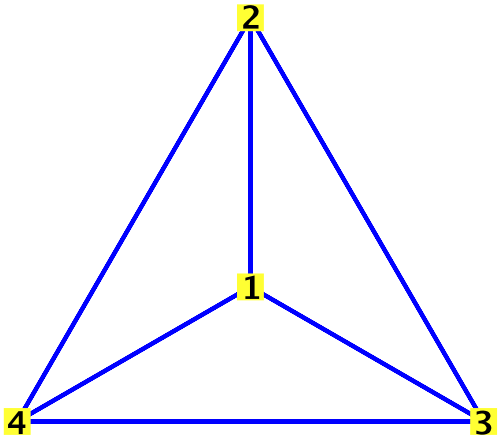
```
> A := Array(0..2,1..2,[[1,2],[3,4],[5,6]]);
```

$$A := Array(0..2, 1..2, \{(0, 1) = 1, (0, 2) = 2, (1, 1) = 3, (1, 2) = 4, (2, 1) = 5, (2, 2) = 6\},$$
$$datatype = anything, storage = rectangular, order = Fortran\_order)$$

```
> ?Array
> with(GraphTheory):
> K4 := CompleteGraph(4);
```

$$K4 := Graph\ 2: an\ undirected\ unweighted\ graph\ with\ 4\ vertices\ and\ 6\ edge(s)$$

```
> DrawGraph(K4);
```




```
> Vertices(K4);
```

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

```
> Edges(K4);
```

$$\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$$

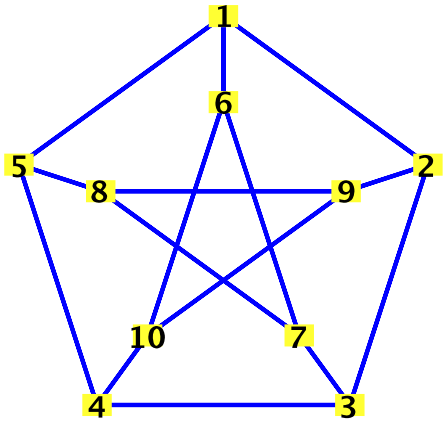
```
> Neighbors(K4);
```

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

```
> with(SpecialGraphs):
> P := PetersenGraph();
```

$$P := Graph\ 3: an\ undirected\ unweighted\ graph\ with\ 10\ vertices\ and\ 15\ edge(s)$$

```
> DrawGraph(P);
```




```
> IsPlanar(P);
```

false

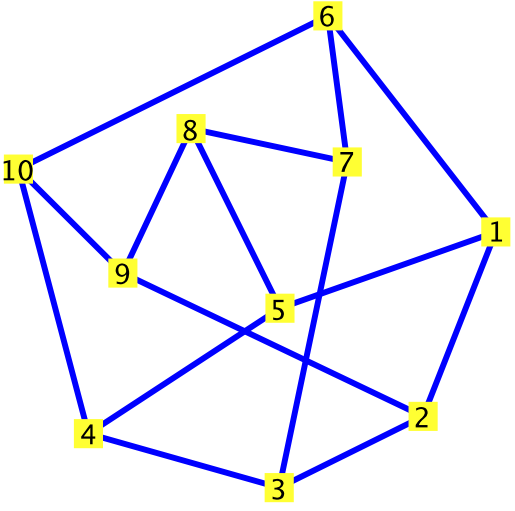
```
> IsPlanar(K4);
```

true

```
> TuttePolynomial(P,u,v);
```

$$u^9+6\,u^8+21\,u^7+56\,u^6+12\,u^5\,v+v^6+114\,u^5+70\,u^4\,v+30\,u^3\,v^2+15\,u^2\,v^3+10\,u\,v^4+9\,v^5 +170\,u^4+170\,u^3\,v+105\,u^2\,v^2+65\,u\,v^3+35\,v^4+180\,u^3+240\,u^2\,v+171\,u\,v^2+75\,v^3 +120\,u^2+168\,u\,v+84\,v^2+36\,u+36\,v$$

```
> DrawGraph(P,style=spring);
```




```
> DrawGraph(P,style=spring,animate=true):
```