

> **with (GraphTheory) ;**

[*AcyclicPolynomial, AddArc, AddEdge, AddVertex, AdjacencyMatrix, AllPairsDistance, Arrivals, ArticulationPoints, AutomorphismGroup, BellmanFordAlgorithm, BiconnectedComponents, BipartiteMatching, Blocks, CanonicalGraph, CartesianProduct, CharacteristicPolynomial, ChromaticIndex, ChromaticNumber, ChromaticPolynomial, CircularChromaticIndex, CircularChromaticNumber, CircularEdgeChromaticNumber, CliqueCover, CliqueCoverNumber, CliqueNumber, CliquePolynomial, CompleteGraph, ConnectedComponents, Contract, ConvertGraph, CopyGraph, CycleBasis, CycleGraph, Degree, DegreeSequence, DelaunayTriangulation, DeleteArc, DeleteEdge, DeleteVertex, Departures, Diameter, Digraph, DijkstrasAlgorithm, DiscardEdgeAttribute, DiscardGraphAttribute, DiscardVertexAttribute, DisjointUnion, Distance, DistancePolynomial, DrawAutomorphism, DrawGraph, DrawNetwork, DrawPlanar, Eccentricity, EdgeChromaticNumber, EdgeConnectivity, Edges, ExportGraph, FindClique, FindHamiltonianCycle, FindHamiltonianPath, FindVertexCover, FlowPolynomial, FundamentalCycle, GetEdgeAttribute, GetEdgeWeight, GetGraphAttribute, GetVertexAttribute, GetVertexPositions, Girth, GlobalClusteringCoefficient, Graph, GraphComplement, GraphEqual, GraphIntersection, GraphJoin, GraphNormal, GraphPolynomial, GraphPower, GraphRank, GraphSpectrum, GraphUnion, GreedyClique, GreedyColor, GreedyIndependentSet, HasArc, HasEdge, HighlightEdges, HighlightSubgraph, HighlightTrail, HighlightVertex, HighlightedEdges, HighlightedVertices, ImportGraph, InDegree, IncidenceMatrix, IncidentEdges, IndependenceNumber, IndependencePolynomial, InducedSubgraph, IntervalGraph, IsAcyclic, IsAntiArborescence, IsArborescence, IsBiconnected, IsBipartite, IsClique, IsConnected, IsCutSet, IsDirected, IsEdgeColorable, IsEulerian, IsForest, IsGraphicSequence, IsHamiltonian, IsIntegerGraph, IsIsomorphic, IsNetwork, IsPlanar, IsReachable, IsRegular, IsStronglyConnected, IsStronglyRegular, IsTournament, IsTree, IsTriangleFree, IsTwoEdgeConnected, IsVertexColorable, IsWeighted, IsomorphicCopy, KruskalsAlgorithm, LaplacianMatrix, Latex, LineGraph, ListEdgeAttributes, ListGraphAttributes, ListVertexAttributes, LocalClusteringCoefficient, MakeDirected, MakeWeighted, MaxFlow, MaximumClique, MaximumDegree, MaximumIndependentSet, MaximumMatching, MinimalSpanningTree, MinimumDegree, MinimumVertexCover, Mycielski, Neighborhood, Neighbors, NonIsomorphicGraphs, NumberOfEdges, NumberOfSpanningTrees, NumberOfVertices, OddGirth, OutDegree, PathGraph, PermuteVertices, PlaneDual, PrimsAlgorithm, Radius, RandomGraphs, RankPolynomial, Reachable, RelabelVertices, ReliabilityPolynomial, ReverseGraph, SeidelSpectrum, SeidelSwitch, SequenceGraph, SetEdgeAttribute, SetEdgeWeight, SetGraphAttribute, SetVertexAttribute, SetVertexPositions, ShortestPath, SpanningPolynomial, SpanningTree, SpecialGraphs, StronglyConnectedComponents, Subdivide, Subgraph, TensorProduct, TopologicSort, Trail, TransitiveClosure, TransitiveReduction, TravelingSalesman, TreeHeight, TuttePolynomial, TwoEdgeConnectedComponents, UnderlyingGraph, VertexConnectivity, VertexCoverNumber, Vertices, WeightMatrix]*

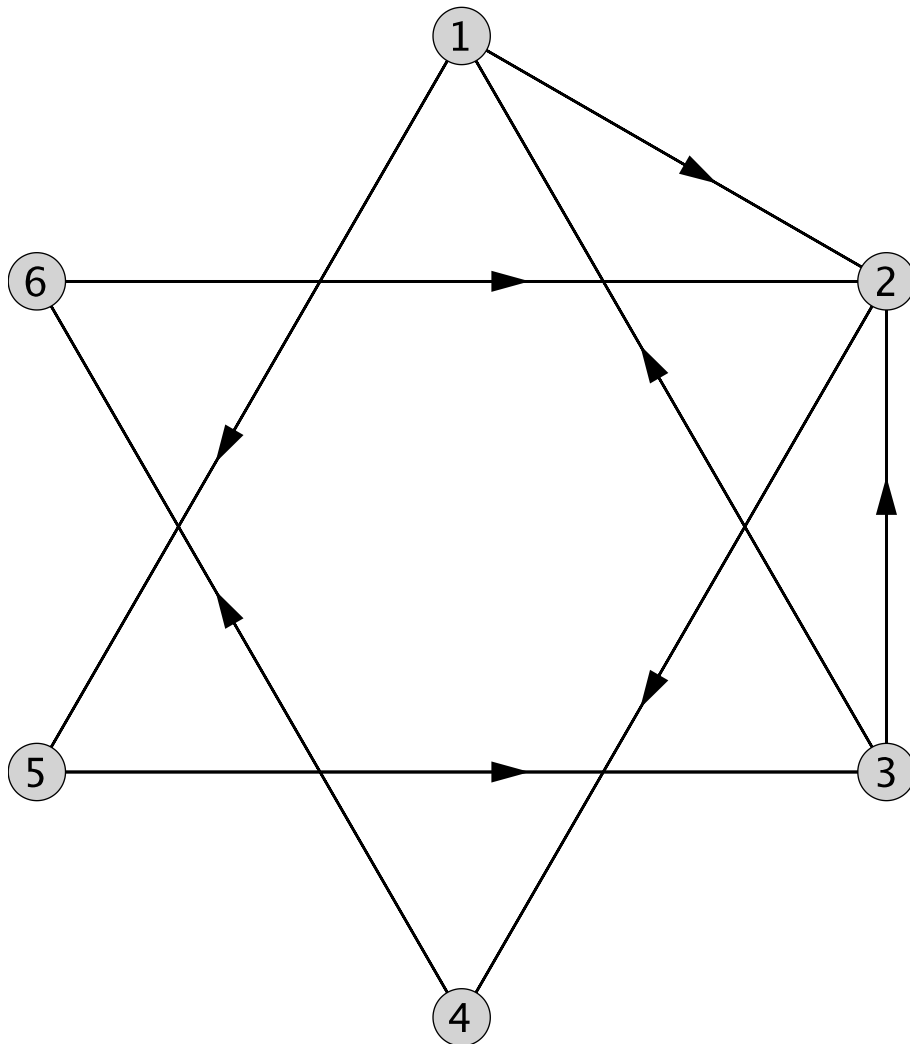
> **Dg := Digraph([1,2,3,4,5,6], {[1,2],[1,5],[2,4],[3,1],[3,2],[4,6],[5,3],[6,2]});**

(1)

Dg := Graph 3: a directed unweighted graph with 6 vertices and 8 arc(s)

(2)

> DrawGraph (Dg) ;



> M := AdjacencyMatrix (Dg) ;

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

(3)

> M[2,2]:=1: M[3,3]:=1: M[6,6]:=1: M;

(4)

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

(4)

> M^2;

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

(5)

> M^3;

$$\begin{bmatrix} 1 & 2 & 1 & 1 & 0 & 1 \\ 0 & 2 & 0 & 1 & 0 & 2 \\ 1 & 5 & 2 & 3 & 1 & 1 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 3 & 1 & 1 & 1 & 0 \\ 0 & 3 & 0 & 2 & 0 & 2 \end{bmatrix}$$

(6)

> with (LinearAlgebra) ;

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, ProjectionMatrix,

(7)

QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

```
> S3 := IdentityMatrix(6) + M + M^2 + M^3;
```

$$S3 := \begin{bmatrix} 2 & 4 & 2 & 2 & 1 & 1 \\ 0 & 5 & 0 & 3 & 0 & 3 \\ 3 & 9 & 5 & 4 & 2 & 1 \\ 0 & 3 & 0 & 2 & 0 & 3 \\ 2 & 4 & 3 & 1 & 2 & 0 \\ 0 & 6 & 0 & 3 & 0 & 5 \end{bmatrix}$$

(8)

```
> map(signum, S3);
```

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

(9)

```
> R1 := map(signum, S3 + M^4 + M^5);
```

$$R1 := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(10)

```
> R2 := map(signum, (IdentityMatrix(6) + M)^(2^3));
```

$$R2 := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(11)

```
> R1 - R2;
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(12)

> R1^2;

$$\begin{bmatrix} 3 & 6 & 3 & 6 & 3 & 6 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 3 & 6 & 3 & 6 & 3 & 6 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 3 & 6 & 3 & 6 & 3 & 6 \\ 0 & 3 & 0 & 3 & 0 & 3 \end{bmatrix}$$

(13)

> R1 *~ Transpose(R1);

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

(14)

```
> for x from -1 to 1 do x, "sign=", sign(x), "signum=", signum(x); od;
-1, "sign=", -1, "signum=", -1
0, "sign=", 1, "signum=", 0
1, "sign=", 1, "signum=", 1
```

(15)

>