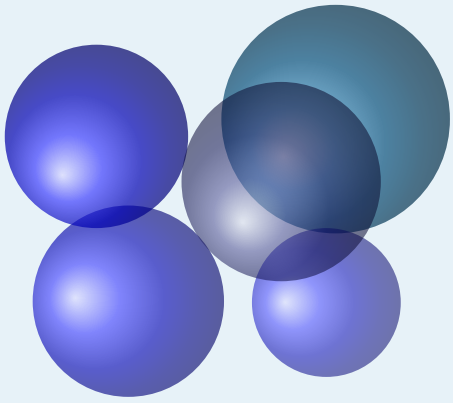
tkz-berge v 2.30c

AlterMundus



Alain Matthes

2 novembre 2026 Documentation V.

http://altermundus.fr

tkz-berge

Alain Matthes

AlterMundus

The package **tkz-berge** is a collection of some useful macros if you want to draw some classic graphs of the graph theory or to make others graphs. The kind of graphs that I will present, are sometimes called combinatorial graphs to distinguish them from the graphs of functions. Often, the word graph is short for graph of a function. A combinatorial graph is a very simple structure, a bunch of dots, some of which are connected by lines. Some of graphs have names, sometimes inspired by the graph's topology, and sometimes after their discoverer.

Why tkz-berge.sty?

Claude Berge (1926 – 2002) was a French mathematician, recognized as one of the modern founders of combinatorics and graph theory. He played a major role in the renaissance of combinatorics and he is remembered for his famous conjecture on perfect graphs, solved some months after his death.

☞ Firstly, I would like to thank **Till Tantau** for the beautiful LATEX package, namely TikZ.
☞I am grateful to **Michel Bovani** for providing the **fourier** font.
☞ I received much valuable advice and guidance on Graph Theory from **Rafael Villarroel** http://graphtheoryinlatex.blogspot.com/.
☞ The names of graphs can be found here MathWorld - SimpleGraphs by E.Weisstein

Please report typos or any other comments to this documentation to Alain Matthes This file can be redistributed and/or modified under the terms of the LATEX Project Public License Distributed from CTAN archives in directory CTAN://macros/latex/base/lppl.txt.

Table des matières

1 Macros and Vertices 7
1.1 \grEmptyCycle . . . 7
1.1.1 Empty Cycle . . . 7
1.1.2 Empty Cycle and \SetVertexNoLabel . . . 7
1.1.3 Empty Cycle and Math . . . 7
1.1.4 Empty Cycle, \SetVertexMath and prefix . . . 8
1.1.5 Empty Cycle and Classic style . . . 8
1.1.6 Empty Cycle and Simple style . . . 8
1.2 \grEmptyPath . . . 9
1.2.1 Empty Path, RA and Math . . . 9
1.2.2 Empty Path, RA and prefix . . . 9
1.2.3 Empty Path, vertical path with form=2 . . . 9
1.2.4 Two Empty Paths . . . 10
1.2.5 How to move a graph? . . . 11

2 Macros and Edges in a graph 12
2.1 Edge in a graph from one vertex \EdgeInGraphFromOneToComp . . . 12
2.1.1 Empty Cycle . . . 12
2.2 Edges in a graph - a loop \EdgeInGraphLoop . . . 13
2.2.1 Empty Cycle . . . 13
2.2.2 Empty Cycle . . . 13
2.3 Edges in a graph - a loop \EdgeInGraphLoop* . . . 14
2.3.1 Empty Cycle . . . 14
2.3.2 Empty Path . . . 14
2.4 Sequence of edges in a graph \EdgeInGraphSeq . . . 15
2.4.1 EdgeInGraphSeq . . . 15
2.5 Edges in a graph \EdgeInGraphMod . . . 16
2.5.1 EdgeInGraphMod . . . 16
2.5.2 EdgeInGraphMod 2 . . . 16
2.6 Edges in a graph \EdgeInGraphMod* . . . 17
2.6.1 EdgeInGraphMod* . . . 17
2.7 Edges in a graph \EdgeInGraphModLoop . . . 18
2.7.1 EdgeInGraphModLoop . . . 18
2.7.2 EdgeInGraphModLoop . . . 19
2.8 Edges between two graphs with the same order \EdgeIdentity . . . 20
2.8.1 EdgeIdentity . . . 20
2.9 Edges between two graphs with the same order \EdgeIdentity* . . . 21
2.9.1 EdgeIdentity* . . . 21
2.9.2 EdgeIdentity* . . . 22
2.10 Edges between two graphs \EdgeFromOneToAll . . . 23
2.10.1 EdgeFromOneToAll . . . 23
2.11 Edges between two graphs \EdgeFromOneToSeq . . . 24
2.11.1 EdgeFromOneToSeq . . . 24
2.12 Edges between two graphs \EdgeFromOneToSel . . . 25
2.12.1 EdgeFromOneToSel . . . 25
2.13 Edges between two graphs \EdgeFromOneToComp . . . 26
2.13.1 EdgeFromOneToComp . . . 26
2.14 Edges between two graphs \EdgeMod . . . 27
2.14.1 EdgeMod . . . 27
2.15 Edges between two graphs \EdgeMod* . . . 28
2.15.1 \EdgeMod* . . . 28

2.15.2 EdgeMod* . . . 29
2.16 Edges between two graphs \EdgeDoubleMod . . . 30
2.16.1 EdgeDoubleMod . . . 30
2.16.2 EdgeDoubleMod with two graphs and different orders . . . 31

3 Classic Graphs 32
3.0.1 Cycle graph . . . 32
3.0.2 Special cases : the triangle graph and the square graph . . . 32
3.0.3 Complete graph . . . 33
3.0.4 Complete Graph order 4 . . . 33
3.0.5 Complete Graph order 4 . . . 34
3.0.6 Circulant graph . . . 35
3.0.7 Graph order 5 with L={1} . . . 35
3.0.8 Graph order 5 with L={2} . . . 36
3.0.9 Graph order 5 with L={1,2} . . . 36
3.0.10 Graph order 10 with L={1,2,3,4,5} . . . 36
3.0.11 Graph order 10 with L={3} . . . 37
3.0.12 Graph order 21 with L={1,3,10} . . . 38
3.0.13 Star graph . . . 39
3.0.14 Star graph . . . 39
3.0.15 Square graph . . . 40
3.0.16 Square Cycle graph . . . 40
3.0.17 Wheel graph . . . 41
3.0.18 Wheel graph . . . 41
3.0.19 Ladder graph . . . 42
3.0.20 Ladder graph . . . 42
3.0.21 Prism graph . . . 43
3.0.22 Cycle Ladder graph . . . 43
3.0.23 Cycle Ladder graph number 3 . . . 44
3.0.24 Cycle Ladder graph number 4 . . . 44
3.0.25 Complete Bipartite graph . . . 45
3.0.26 Bipartite graph 1,5 . . . 45
3.0.27 Bipartite graph 3,5 . . . 46
3.0.28 Triangular Grid graph . . . 47
3.0.29 n=8 order=28 form 1 . . . 47
3.0.30 n=6 order=15 form 2 . . . 48
3.0.31 n=6 order=15 form 3 . . . 49
3.0.32 LCF Lederberg-Coxeter-Fruchte . . . 50
3.0.33 $[2,-2]^2$ . . . 50
3.0.34 $[3,-3]^4$ . . . 51
3.0.35 Ljubljana graph . . . 51

4 New graphs and a gallery 53
4.1 Hypercubes : \grHypercube . . . 53
4.2 Trees : \grTree . . . 53
4.3 A gallery . . . 55
4.3.1 The Petersen graph is a Kneser graph . . . 56
4.3.2 Ramsey : $R(3,3) = 6$ . . . 57
4.3.3 The Moser spindle . . . 58
4.3.4 Mycielski's construction . . . 59
4.3.5 The de Bruijn graph $B(2,3)$ . . . 60
4.3.6 The Paley graph P_{13} . . . 61
4.3.7 The hypercube Q_4 and the subsets of a set . . . 62

4.3.8 The Tower of Hanoi 64
4.3.9 The Hoffman–Singleton graph 65
4.3.10 The mystic rose 67
4.3.11 The Coxeter graph 68

5 Macros and Styles 69
5.1 How to change the background color and text color 69
5.2 Modification of labels **\AssignVertexLabel** 69
5.2.1 AssignStyle and **\AssignVertexLabel** 69

Index 71

List of the main macros :

- \grEmptyCycle
- \grEmptyPath
- \grEmptyStar
- \grEmptyGrid
- \grEmptyLadder
- \EdgeInGraphFromOneToComp
- \EdgeInGraphLoop
- \EdgeInGraphSeq
- \EdgeInGraphMod
- \EdgeInGraphMod*
- \grCompleteBipartite
- \EdgeInGraphModLoop
- \EdgeIdentity
- \EdgeIdentity*
- \EdgeFromOneToAll
- \EdgeFromOneToSeq
- \EdgeFromOneToSel
- \EdgeFromOneToComp
- \EdgeMod
- \EdgeMod*
- \EdgeDoubleMod
- \grPath
- \grCycle
- \grComplete
- \grCirculant
- \grStar
- \grSQCycle
- \grWheel
- \grLadder
- \grPrism
- \grCompleteBipartite
- \grTriangularGrid
- \grLCF
- \grWriteExplicitLabels
- \grWriteExplicitLabel
- \AssignVertexLabel

Classic graphs :

- \grAndrasfai
- \grBalaban
- \grChvatal
- \grCocktailParty
- \grCrown
- \grCubicalGraph
- \grDesargues
- \grDodecahedral
- \grDoyle
- \grFoster
- \grFolkman
- \grFranklin
- \grGeneralizedPetersen
- \grGrotzsch
- \grHeawood
- \grHypercube
- \grIcosahedral
- \grKonisberg
- \grLevi
- \grMcGee
- \grMobiusKantor
- \grMobiusLadder
- \grOctahedral
- \grPappus
- \grPetersen
- \grRobertson
- \grRobertsonWegner
- \grTetrahedral
- \grTree
- \grTutteCoxeter
- \grWong

See the document "NamedGraph" for all the classic named graphs that you can draw with the package **tkz-berge.sty**.

1 Macros and Vertices

1.1 \grEmptyCycle

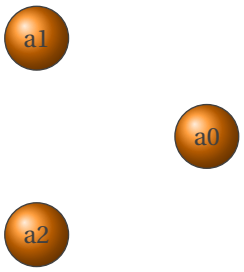
\grEmptyCycle[⟨local options⟩]{⟨order⟩}

Arguments	Definition
order	order of the graph

Options	default	definition
RA	4	radius circle
prefix	a	prefix for vertices
Math	false	math mode

The number of nodes in a graph is called its order. The argument "order" is an integer superior to 1. RA *defines the radius of the circle.*

1.1.1 Empty Cycle



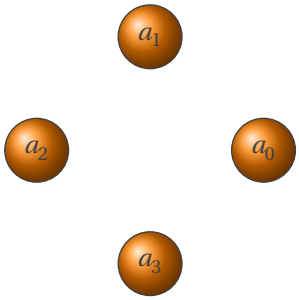

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=1.5]{3}
\end{tikzpicture}
```

1.1.2 Empty Cycle and \SetVertexNoLabel



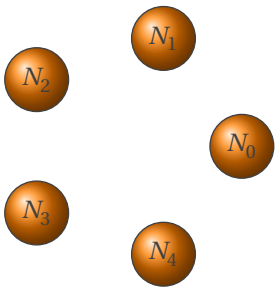
```
\begin{tikzpicture}
  \SetVertexNoLabel
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=1.5]{2}
\end{tikzpicture}
```

1.1.3 Empty Cycle and Math



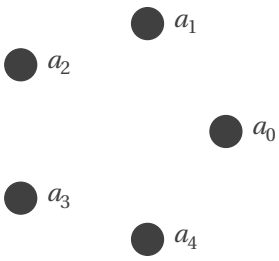

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[Math,RA=1.5]{4}
\end{tikzpicture}
```

1.1.4 Empty Cycle, \SetVertexMath and prefix



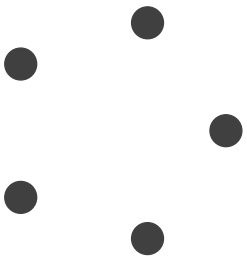

```
\begin{tikzpicture}
  \SetVertexMath
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[prefix=N,RA=1.5]{5}
\end{tikzpicture}
```

1.1.5 Empty Cycle and Classic style




```
\begin{tikzpicture}
  \SetVertexMath
  \GraphInit[vstyle=Classic]
  \grEmptyCycle[RA=1.5]{5}
\end{tikzpicture}
```

1.1.6 Empty Cycle and Simple style



```
\begin{tikzpicture}
  \GraphInit[vstyle=Simple]
  \grEmptyCycle[RA=1.5]{5}
\end{tikzpicture}
```

1.2 \grEmptyPath

\grEmptyPath[⟨local options⟩]{⟨order⟩}

Arguments	Definition
order	order of the graph

options	default	definition
RA	4 cm	distance between two vertices
RS	? cm	distance between the first line and the new one
prefix	a	prefix for vertices
Math	false	math mode

Order *is the number of nodes.* RA *defines the radius of the circle.* RS *defines the distance between the graph and the baseline.*

1.2.1 Empty Path, RA and Math



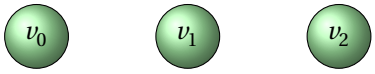

```
\begin{tikzpicture}
   \grEmptyPath[Math,RA=2]{5}
\end{tikzpicture}
```

1.2.2 Empty Path, RA and prefix




```
\begin{tikzpicture}
   \grEmptyPath[prefix=h,RA=2]{6}
\end{tikzpicture}
```

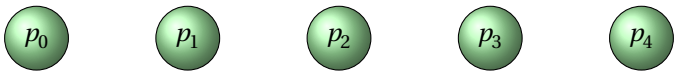
1.2.3 Empty Path, vertical path with form=2




```
\begin{tikzpicture}
 \grEmptyPath[form=2,prefix=v,RA=2]{3}
\end{tikzpicture}
```

1.2.4 Two Empty Paths





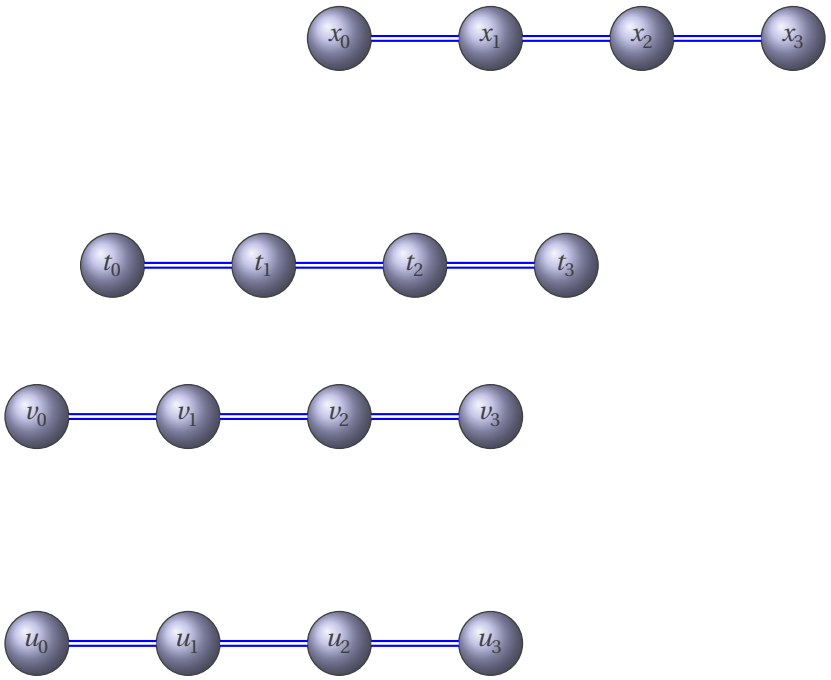

```
\begin{tikzpicture}
    \grEmptyPath[Math,prefix=p,RA=2,RS=0]{5}
    \grEmptyPath[Math,prefix=q,RA=2,RS=3]{5}
\end{tikzpicture}
```






```
\begin{tikzpicture}
    \grEmptyPath[Math,prefix=p,RA=2,RS=0,form=2]{5}
    \grEmptyPath[Math,prefix=q,RA=2,RS=4,form=2]{5}
\end{tikzpicture}
```

1.2.5 How to move a graph ?




```
\begin{tikzpicture}
     \GraphInit[vstyle=Shade]
     \SetGraphShadeColor{blue!60!black!30}{blue}{white}
     \grPath[Math,prefix=u,RA=2,RS=0]{4}
     \grPath[Math,prefix=v,RA=2,RS=3]{4}
     \begin{scope}[xshift=1 cm]
      \grPath[Math,prefix=t,RA=2,RS=5]{4}
     \end{scope}
     \begin{scope}[shift={(4 cm,8cm)}]
      \grPath[Math,prefix=x,RA=2,RS=0]{4}
     \end{scope}
\end{tikzpicture}
```

2 Macros and Edges in a graph

2.1 Edge in a graph from one vertex \EdgeInGraphFromOneToComp

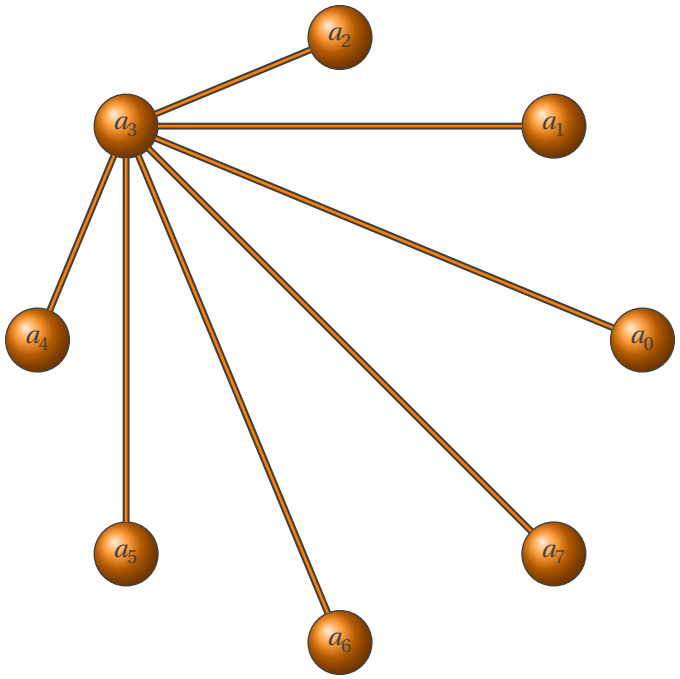
\EdgeInGraphFromOneToComp[⟨local options⟩]{⟨prefix⟩}{⟨order⟩}{⟨from⟩}

Arguments	Definition
order	order of the graph

options	default	definition
RA	4	radius circle
prefix	a	prefix for vertices
Math	false	math mode

This macro works on an unique graph. from *is integer.* EdgeInGraph *designs a macro that works only in a graph defined by a prefix. The result is some edges between the vertex* from *and the others vertices.*

2.1.1 Empty Cycle



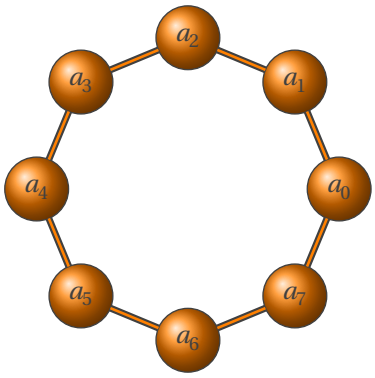

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=4,prefix=a]{8}%
  \EdgeInGraphFromOneToComp{a}{8}{3}
\end{tikzpicture}
```

2.2 Edges in a graph - a loop \EdgeInGraphLoop

\EdgeInGraphLoop{⟨prefix⟩}{⟨order⟩}

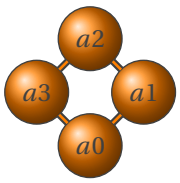
This macro is useful with vertices on a circle . `order` *in an integer.*

2.2.1 Empty Cycle




```
\begin{tikzpicture}
\GraphInit[vstyle=Shade]
\grEmptyCycle[RA=2,prefix=a]{8}%
\EdgeInGraphLoop{a}{8}
\end{tikzpicture}
```

2.2.2 Empty Cycle



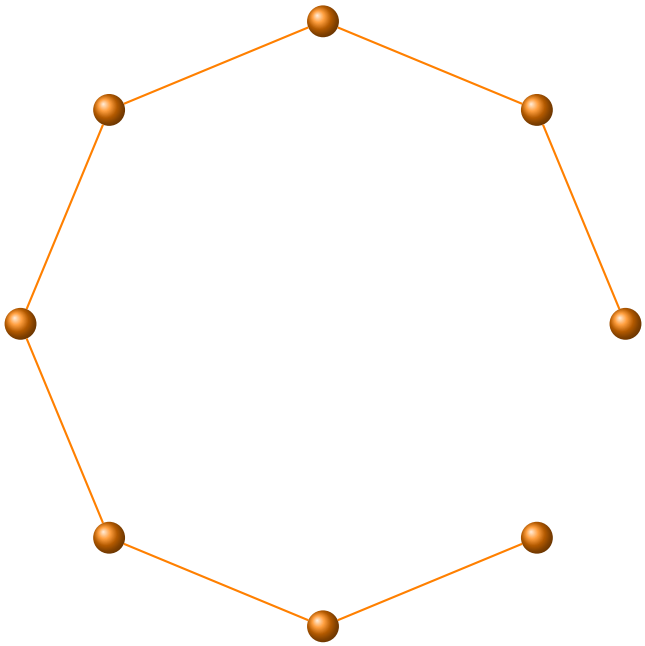

```
\begin{tikzpicture}[node distance=4cm]
 \GraphInit[vstyle=Shade]
 \Vertices{square}{a0,a1,a2,a3}
 \EdgeInGraphLoop{a}{4}
\end{tikzpicture}
```

2.3 Edges in a graph - a loop \EdgeInGraphLoop*

\EdgeInGraphLoop*{⟨prefix⟩}{⟨order⟩}

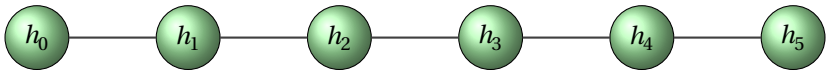
Not exactly a loop, there is no edge between the first and the last vertex.

2.3.1 Empty Cycle



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \grEmptyCycle[RA=4,prefix=a]{8}%
  \EdgeInGraphLoop*{a}{8}
\end{tikzpicture}
```

2.3.2 Empty Path



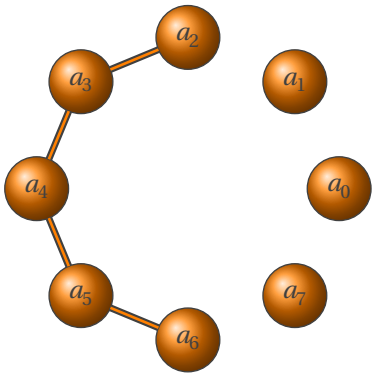

```
\begin{tikzpicture}
     \grEmptyPath[prefix=h,RA=2,RS=2]{6}
     \EdgeInGraphLoop*{h}{6}
\end{tikzpicture}
```

2.4 Sequence of edges in a graph \EdgeInGraphSeq

\EdgeInGraphSeq{⟨prefix⟩}{⟨start⟩}{⟨end⟩}

This macro gives a sequence of edges between start *and* end.
start *and* end *are two integers.*

2.4.1 EdgeInGraphSeq



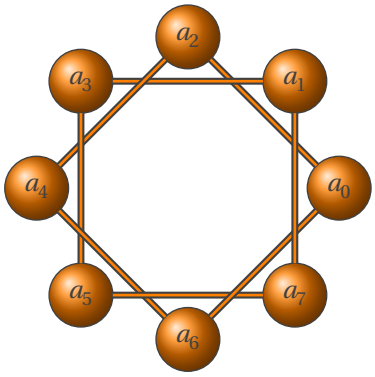

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=2,prefix=a]{8}%
  \EdgeInGraphSeq{a}{2}{5}
\end{tikzpicture}
```

2.5 Edges in a graph \EdgeInGraphMod

\EdgeInGraphMod{⟨prefix⟩}{⟨order⟩}{⟨add⟩}

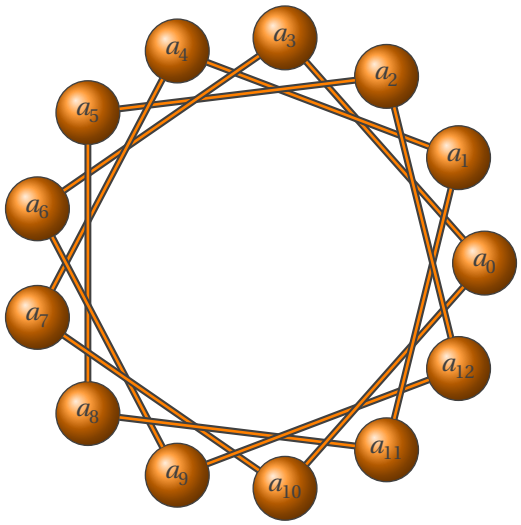
This macro works on an unique graph. Edges between v_i and v_j with i in $0,...,(\#2-1)$ and $j = Mod(i+\#3,\#2)$.
#2 = order *and #3 =* add.
Mod *is like* mod *but the result is a positive integer.*

2.5.1 EdgeInGraphMod




```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=2,prefix=a]{8}%
  \EdgeInGraphMod{a}{8}{2}
\end{tikzpicture}
```

2.5.2 EdgeInGraphMod 2



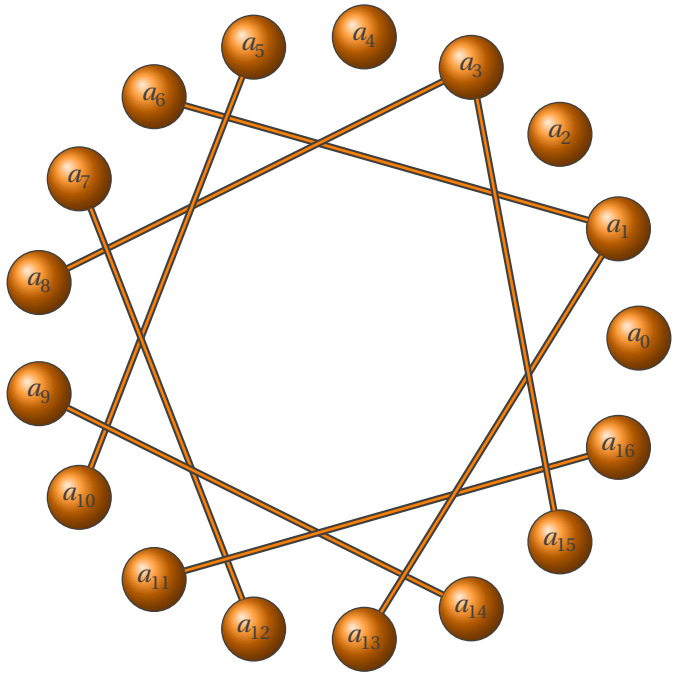

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=3,prefix=a]{13}%
  \EdgeInGraphMod{a}{13}{3}
\end{tikzpicture}
```

2.6 Edges in a graph \EdgeInGraphMod*

\EdgeInGraphMod*{⟨prefix⟩}{⟨order⟩}{⟨add⟩}{⟨start⟩}{⟨step⟩}

Edges between v_i and v_j with i in #4, #4 + #5, ..., (#2 − 1) and j = Mod(i+#3,#2)
#2 = order, #3 = add, #4 = start, #5 = step.

2.6.1 EdgeInGraphMod*



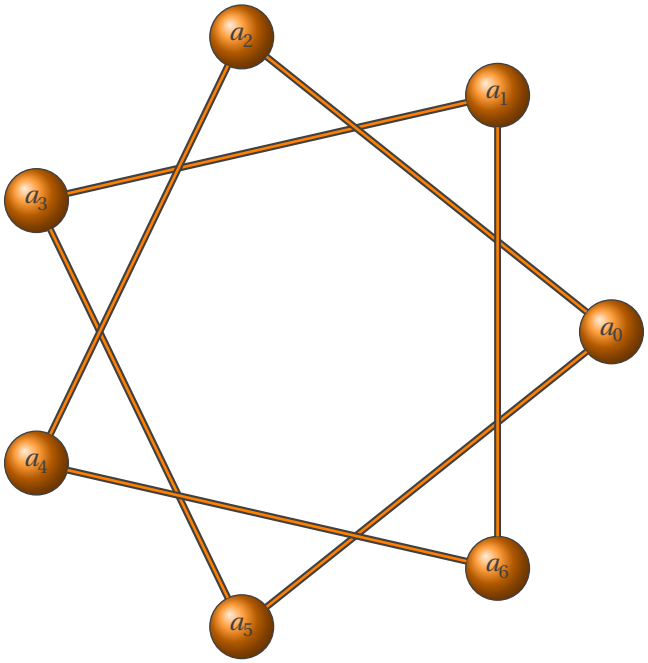

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[prefix=a]{17}%
  \EdgeInGraphMod*{a}{17}{5}{1}{2}
\end{tikzpicture}
```

2.7 Edges in a graph \EdgeInGraphModLoop

\EdgeInGraphModLoop{⟨prefix⟩}{⟨order⟩}{⟨add⟩}{⟨start⟩}

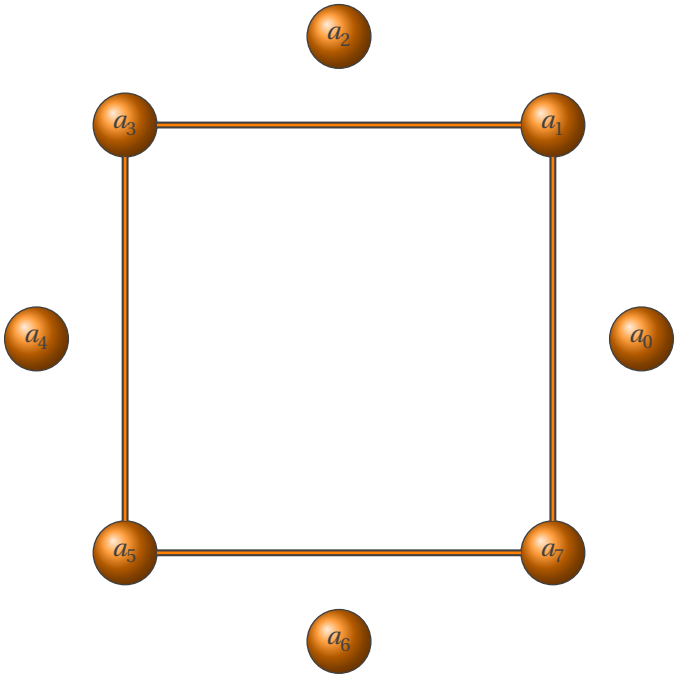
order, add *and* start *are integers.*
Edges between v_i *and* v_j *with* i *from* #4, j = *Mod(i+#3,#2) and then* $i = j$ *until* j = #4
#2 = order, *#3* = add *and #4* = start.

2.7.1 EdgeInGraphModLoop




```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=4]{7}
  \EdgeInGraphModLoop{a}{7}{2}{1}
\end{tikzpicture}
```

2.7.2 EdgeInGraphModLoop



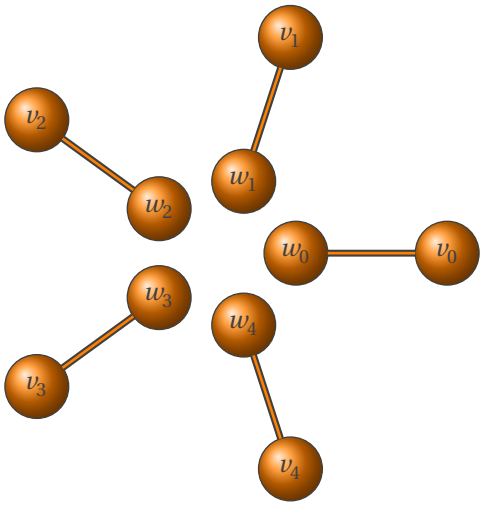

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[RA=4]{8}
  \EdgeInGraphModLoop{a}{8}{2}{1}
\end{tikzpicture}
```

2.8 Edges between two graphs with the same order \EdgeIdentity

\EdgeIdentity{⟨prefix1⟩}{⟨prefix2⟩}{⟨order⟩}

order *is an integer. This macro gives edges between two graphs.*
Edges between v_i *and* v_j *with* $i = j$ *in* $0, ..., (\#3 - 1)$.
#3 = order.

2.8.1 EdgeIdentity



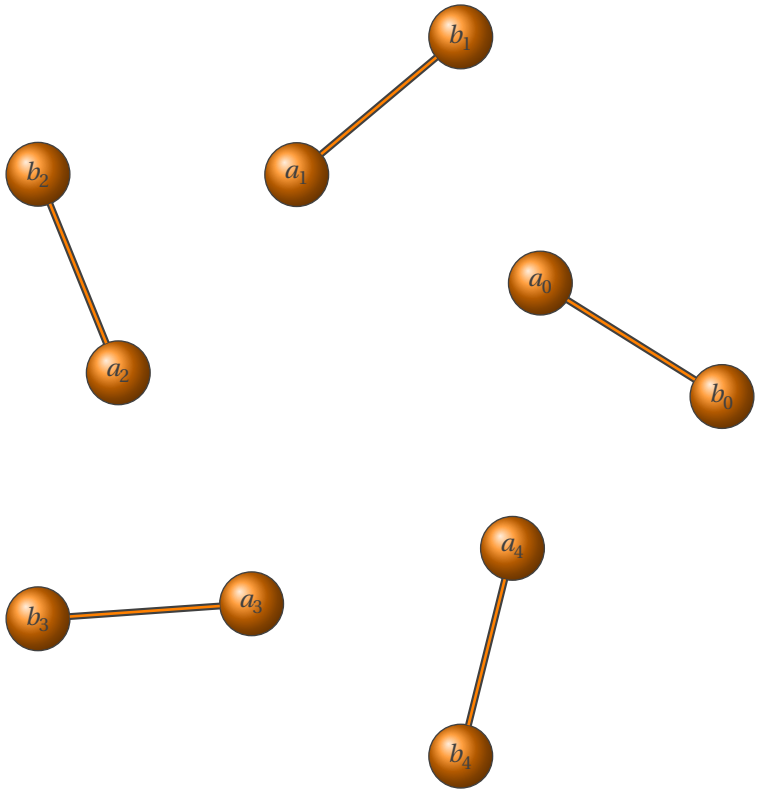

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
    \grEmptyCycle[prefix=v,RA=3]{5}
    \grEmptyCycle[prefix=w,RA=1]{5}
    \EdgeIdentity{v}{w}{5}
\end{tikzpicture}
```

2.9 Edges between two graphs with the same order \EdgeIdentity*

\EdgeIdentity*{⟨prefix1⟩}{⟨prefix2⟩}{⟨list⟩}

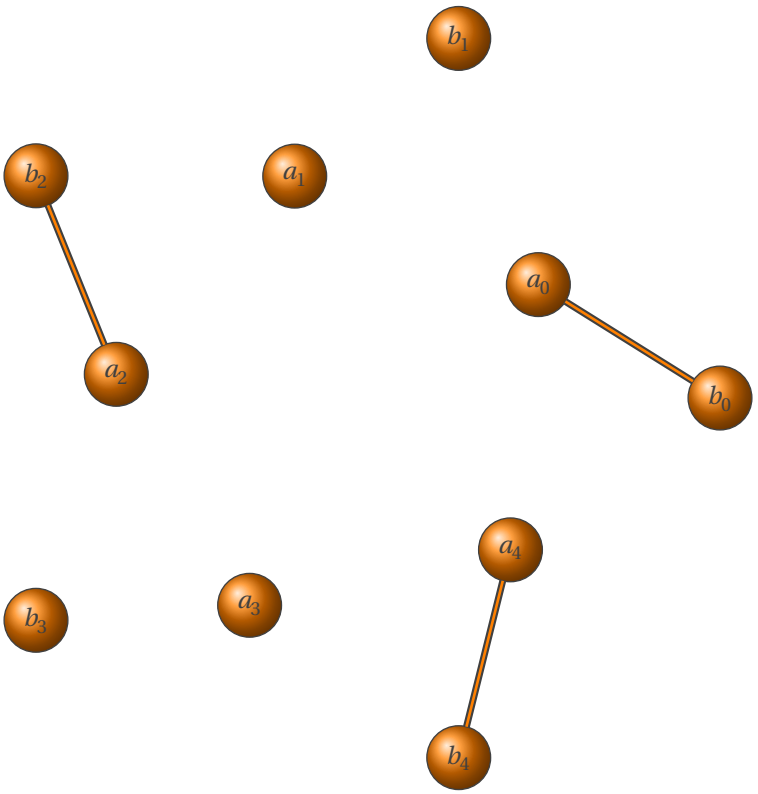
list is a list of integers. This macro gives edges between two graphs.
Edges between v_i and v_j with $i = j$ in list.

2.9.1 EdgeIdentity*




```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \begin{scope}[rotate=30]
  \grEmptyCycle[RA=3,prefix=a]{5}%
  \end{scope}
  \grEmptyCycle[RA=5,prefix=b]{5}%
   \EdgeIdentity*{a}{b}{0,...,4}
\end{tikzpicture}
```

2.9.2 EdgeIdentity*



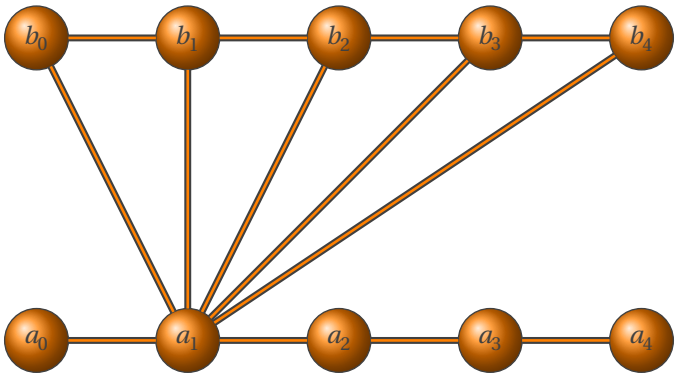

```
\begin{tikzpicture}
 \GraphInit[vstyle=Shade]
  \begin{scope}[rotate=30]
  \grEmptyCycle[RA=3,prefix=a]{5}%
  \end{scope}
  \grEmptyCycle[RA=5,prefix=b]{5}%
   \EdgeIdentity*{a}{b}{0,2,4}
\end{tikzpicture}
```

2.10 Edges between two graphs \EdgeFromOneToAll

\EdgeFromOneToAll{⟨prefix1⟩}{⟨prefix2⟩}{⟨from⟩}{⟨order⟩}

The graphs must to have the same order. `from` *and* `order` *are integers.*

2.10.1 EdgeFromOneToAll



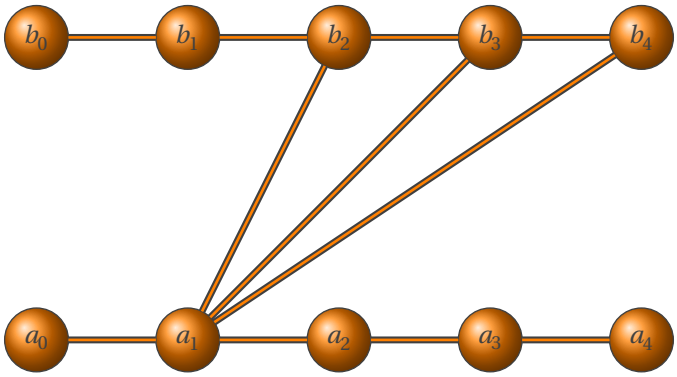

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grPath[form=1,RA=2,RS=0]{5}
  \grPath[form=1,prefix=b,RA=2,RS=4]{5}
  \EdgeFromOneToAll{a}{b}{1}{5}
\end{tikzpicture}
```

2.11 Edges between two graphs \EdgeFromOneToSeq

\EdgeFromOneToSeq{⟨prefix1⟩}{⟨prefix2⟩}{⟨from⟩}{⟨start⟩}{⟨end⟩}

from, start and end are integers. This macro builds edges between the vertex with an indice from through the vertices with an indice in the sequence start,...,end.

2.11.1 EdgeFromOneToSeq



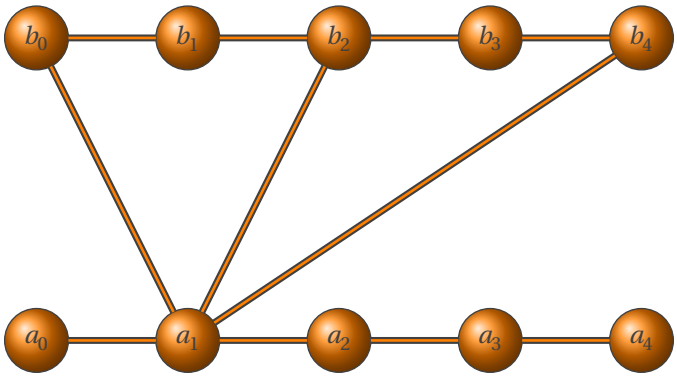

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grPath[form=1,RA=2,RS=9]{5}
  \grPath[form=1,prefix=b,RA=2,RS=4]{5}
  \EdgeFromOneToSeq{a}{b}{1}{2}{4}
\end{tikzpicture}
```

2.12 Edges between two graphs \EdgeFromOneToSel

\EdgeFromOneToSel{⟨prefix1⟩}{⟨prefix2⟩}{⟨from⟩}{⟨list⟩}

This macro builds edges between the vertex with an indice `from` *through the vertices with an indice in the list* `list`.

2.12.1 EdgeFromOneToSel



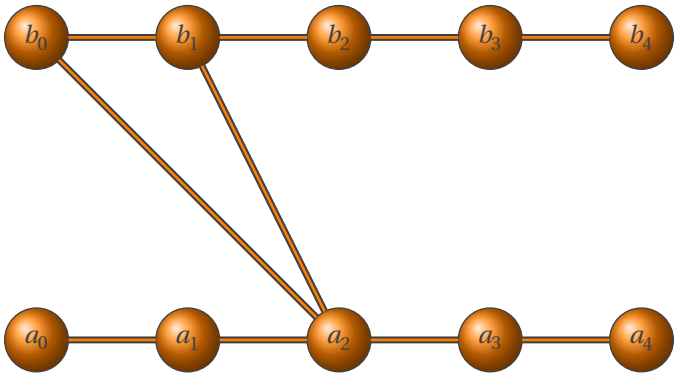

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grPath[form=1,RA=2]{5}
  \grPath[form=1,prefix=b,RA=2,RS=4]{5}
  \EdgeFromOneToSel{a}{b}{1}{0,2,4}
\end{tikzpicture}
```

2.13 Edges between two graphs \EdgeFromOneToComp

\EdgeFromOneToComp{⟨prefix1⟩}{⟨prefix2⟩}{⟨from⟩}{⟨order2⟩}

This macro builds edges between the vertex with an indice `from` *through all the vertices of the second graph, except the vertex with an indice* `from`.

2.13.1 EdgeFromOneToComp



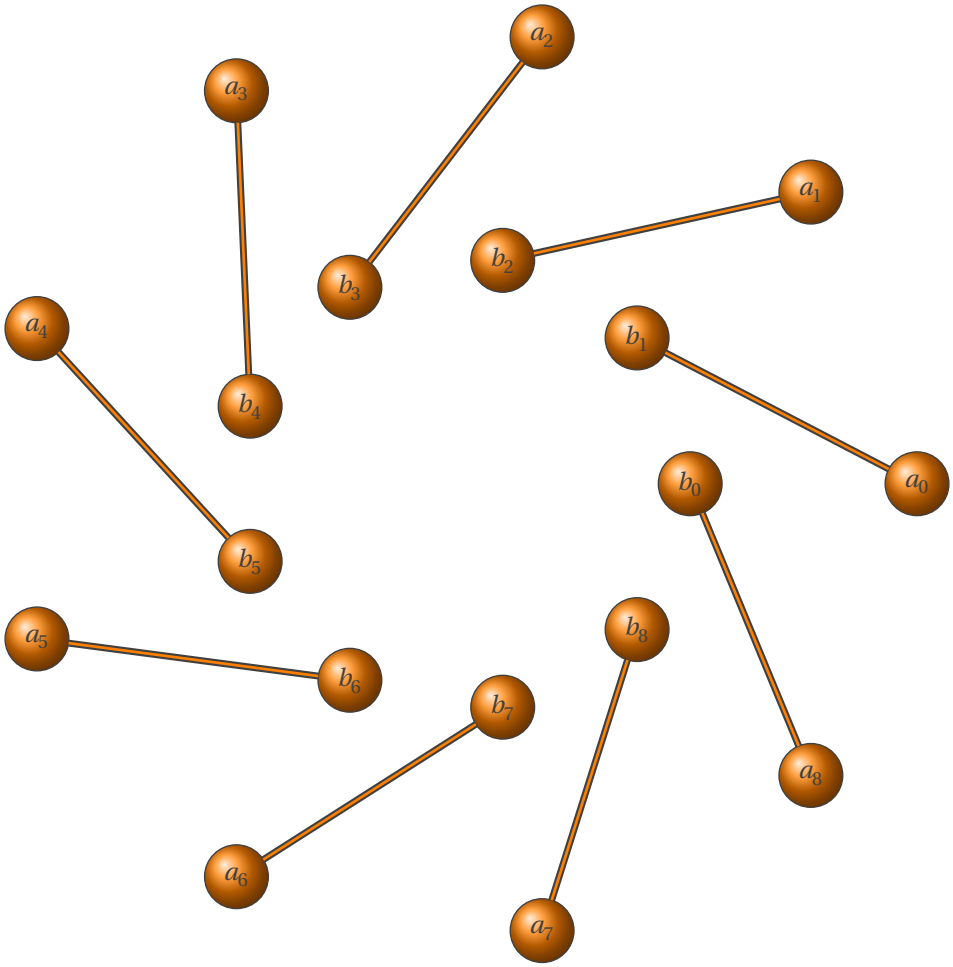

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grPath[form=1,RA=2,RS=0]{5}
  \grPath[form=1,prefix=b,RA=2,RS=4]{5}
  \EdgeFromOneToComp{a}{b}{2}{3}
\end{tikzpicture}
```

2.14 Edges between two graphs \EdgeMod

\EdgeMod{⟨prefix1⟩}{⟨prefix2⟩}{⟨order⟩}{⟨step⟩}

This macro works on two graphs with the same order. We get edges between v_i and v_j with i in $0,...,(\#2-1)$ and $j = Mod(i+\#4,\#3)$.
#3 = order *and #4* = step.

2.14.1 EdgeMod



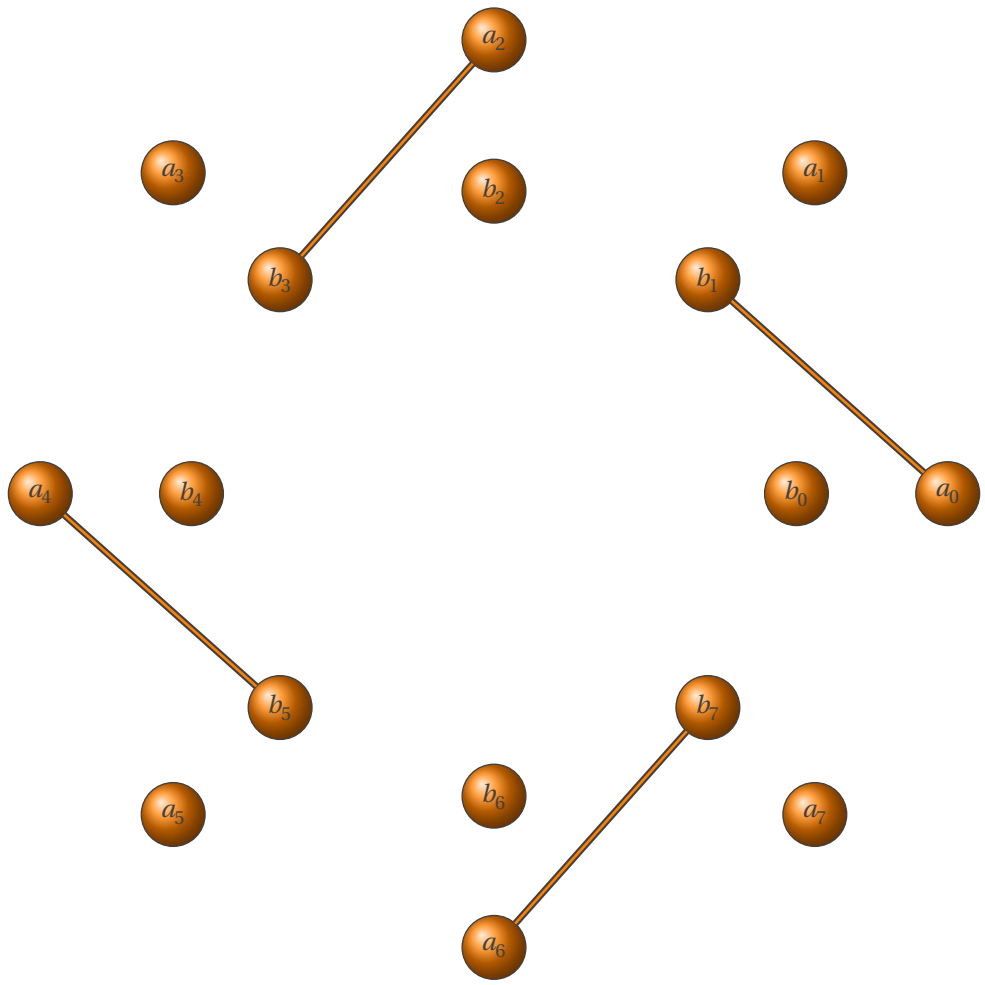

```
\begin{tikzpicture}
 \GraphInit[vstyle=Shade]
 \grEmptyCycle[prefix=a,RA=6]{9}
 \grEmptyCycle[prefix=b,RA=3]{9}
 \EdgeMod{a}{b}{9}{1}
\end{tikzpicture}
```

2.15 Edges between two graphs \EdgeMod*

\EdgeMod*{⟨prefix1⟩}{⟨prefix2⟩}{⟨order⟩}{⟨step1⟩}{⟨step2⟩}

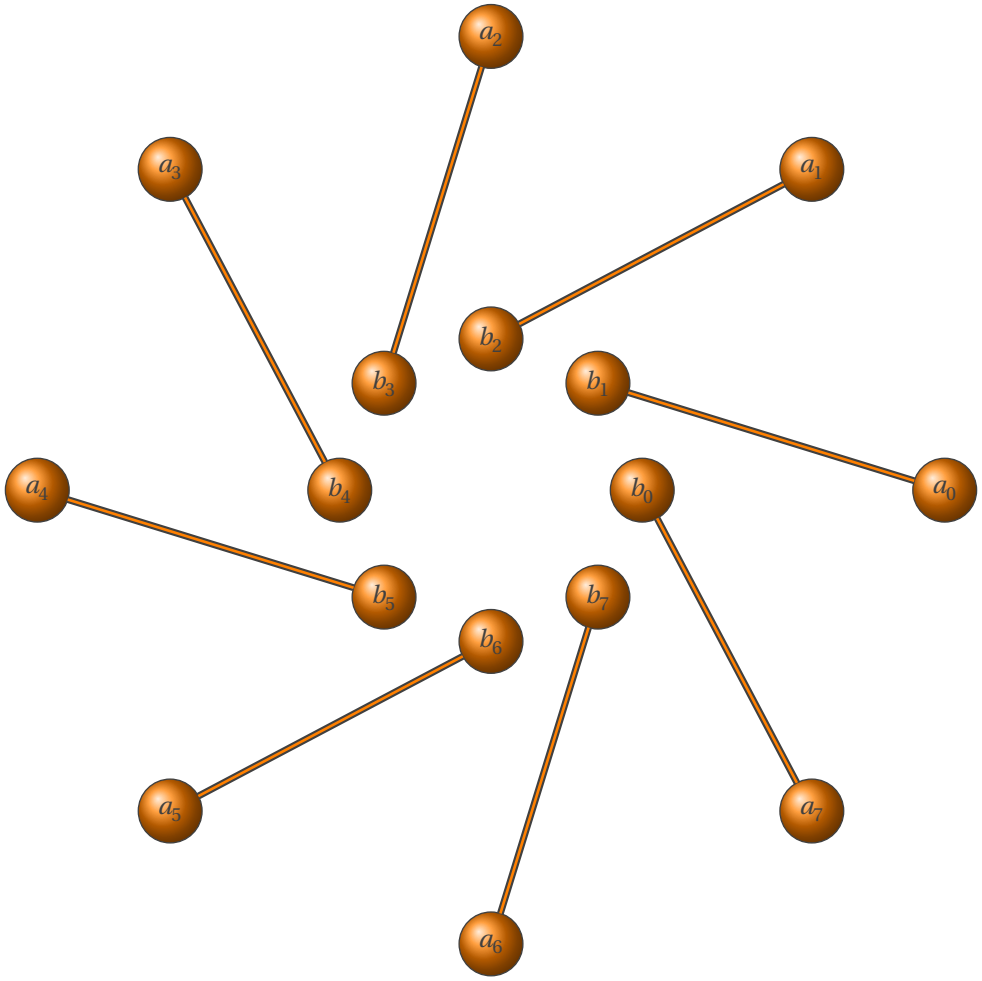
This macro works on two graphs with the same order. We get edges between v_i and v_j with i in $0,...,(\#3-1)$ with a step #5 and j = Mod(i+#4,#3).
#3 = order , *#4* = step1 *and #5* = step2.

2.15.1 \EdgeMod*




```
\begin{tikzpicture}
 \GraphInit[vstyle=Shade]
 \grEmptyCycle[prefix=a,RA=6]{8}
 \grEmptyCycle[prefix=b,RA=4]{8}
 \EdgeMod*{a}{b}{8}{1}{2}
\end{tikzpicture}
```

2.15.2 EdgeMod*




```
\begin{tikzpicture}
 \GraphInit[vstyle=Shade]
 \grEmptyCycle[prefix=a,RA=6]{8}
 \grEmptyCycle[prefix=b,RA=2]{8}
 \EdgeMod*{a}{b}{8}{1}{1}
\end{tikzpicture}
```

2.16 Edges between two graphs \EdgeDoubleMod

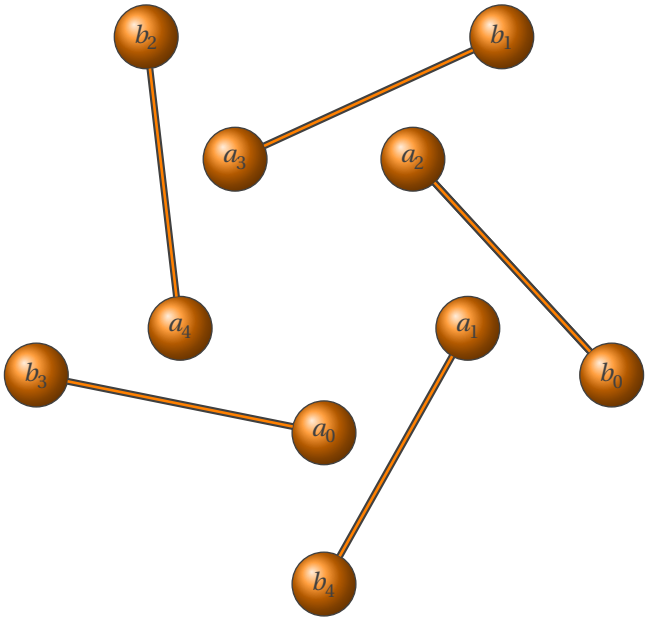
\EdgeDoubleMod{⟨prefix1⟩}{⟨nb⟩}{⟨nb⟩}{⟨nb⟩}{⟨prefix2⟩}{⟨nb⟩}{⟨nb⟩}{⟨nb⟩}{⟨end⟩}

For the first node, the numbers are : {⟨order1⟩}{⟨start1⟩}{⟨add1⟩}

For the second node, the numbers are : {⟨order2⟩}{⟨start2⟩}{⟨add2⟩}{⟨end⟩}

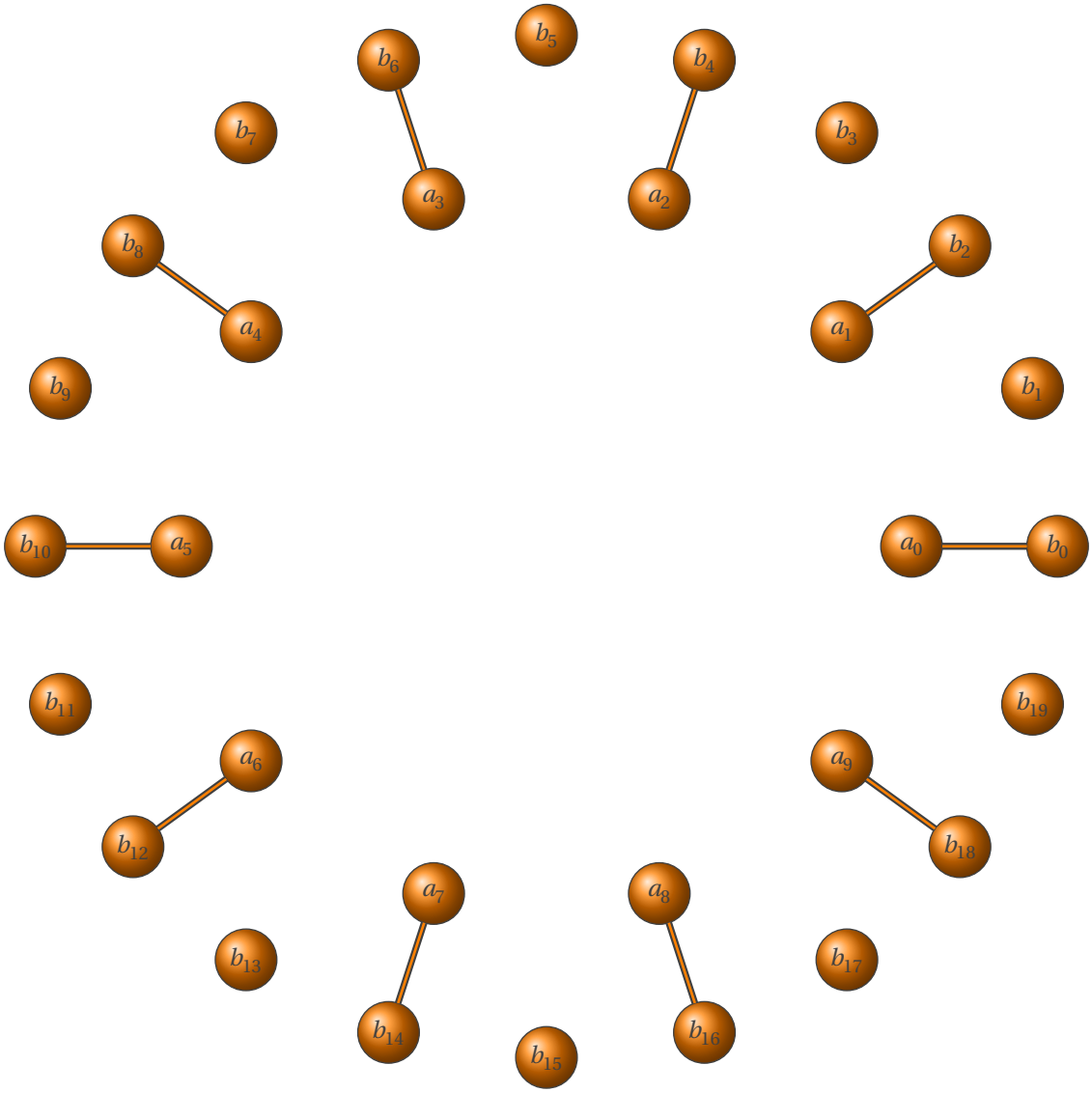
*Edges between v_i and v_j with i = Mod(#3+(#4*k),#2) and j=Mod(#7+(#8*k),#6) k is an integer from* 0 *to* end.
#2 = order1, *#3* = start1 *and #4* = add1.
#6 = order2, *#7* = start2 *and #8* = add2.

2.16.1 EdgeDoubleMod




```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
    \begin{scope}[rotate=-90]
      \grEmptyCycle[RA=2,prefix=a]{5}
    \end{scope}
    \begin{scope}[rotate=-18]
      \grEmptyCycle[RA=4,prefix=b]{5}
    \end{scope}
    \EdgeDoubleMod{b}{5}{0}{1}%
                  {a}{5}{2}{1}{5}
\end{tikzpicture}
```

2.16.2 EdgeDoubleMod with two graphs and different orders




```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \grEmptyCycle[prefix=a,RA=5]{10}
  \grEmptyCycle[prefix=b,RA=7]{20}
   \EdgeDoubleMod{a}{10}{0}{1}%
                 {b}{20}{0}{2}{10}
\end{tikzpicture}
```

3 Classic Graphs

3.0.1 Cycle graph

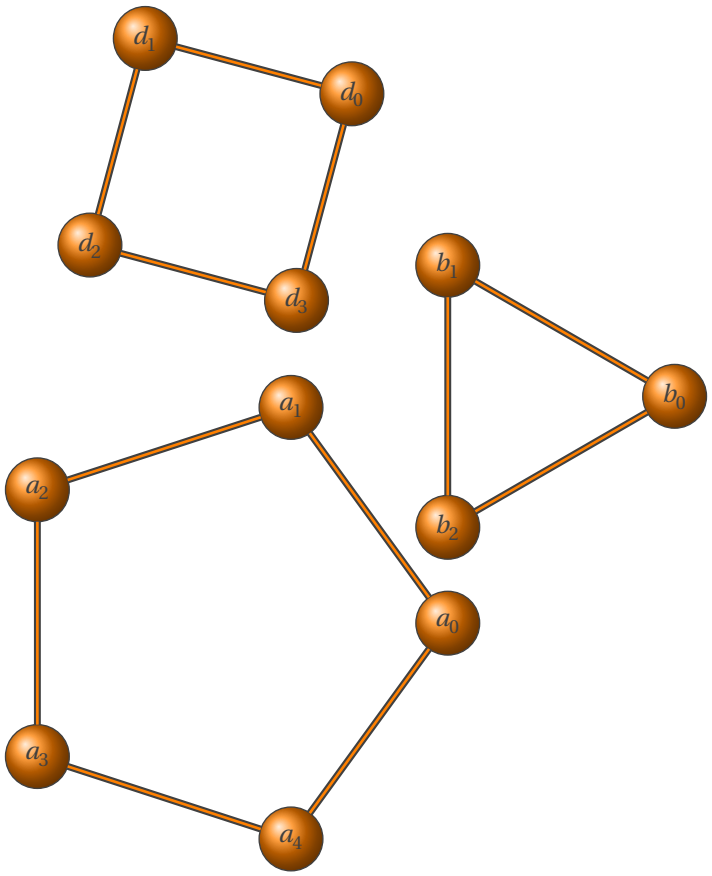
`\grCycle[⟨local options⟩]{⟨order⟩}`

A cycle graph C_n is a graph on n nodes containing a single cycle through all nodes. Cycle graphs can be generated using `\grCycle` *in the* `tkz-berge.sty` *package. Special cases include the triangle graph and the square graph.*

External links :

— MathWorld - CycleGraph by E.Weisstein
— Wikipedia

3.0.2 Special cases : the triangle graph and the square graph




```
\begin{tikzpicture}
    \GraphInit[vstyle=Shade]
    \grCycle[prefix=a,RA=3]{5}
    \grCycle[x=4,y=3,prefix=b,RA=2]{3}
    \grCycle[prefix=d,y=6,rotation=30,RA=2]{4}
\end{tikzpicture}
```

3.0.3 Complete graph

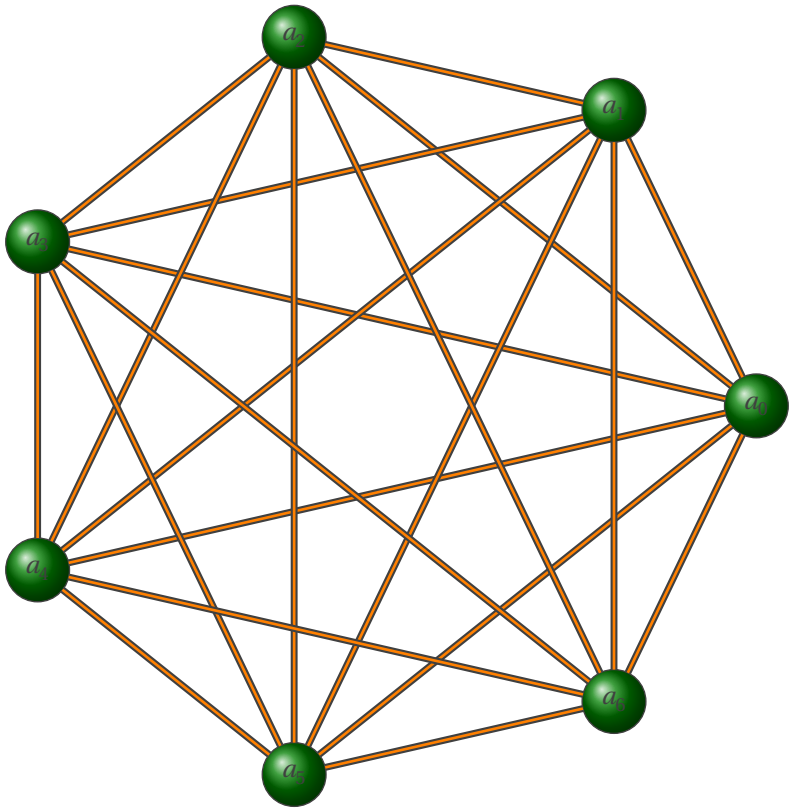
`\grComplete[⟨local options⟩]{⟨order⟩}`

The more simple definition is "an undirected graph with an edge between every pair of vertices" or a complete graph is a simple graph in which each pair of graph vertices is connected by an edge. The complete graph with n graph vertices is denoted K_n. This graph has $\frac{n(n-1)}{2}$ undirected edges.
Geometrically, K_3 relates to a triangle,K_4 a tetrahedron is the tetrahedral graph as well as the wheel graph , K_5 a pentachoron, etc …

External links :

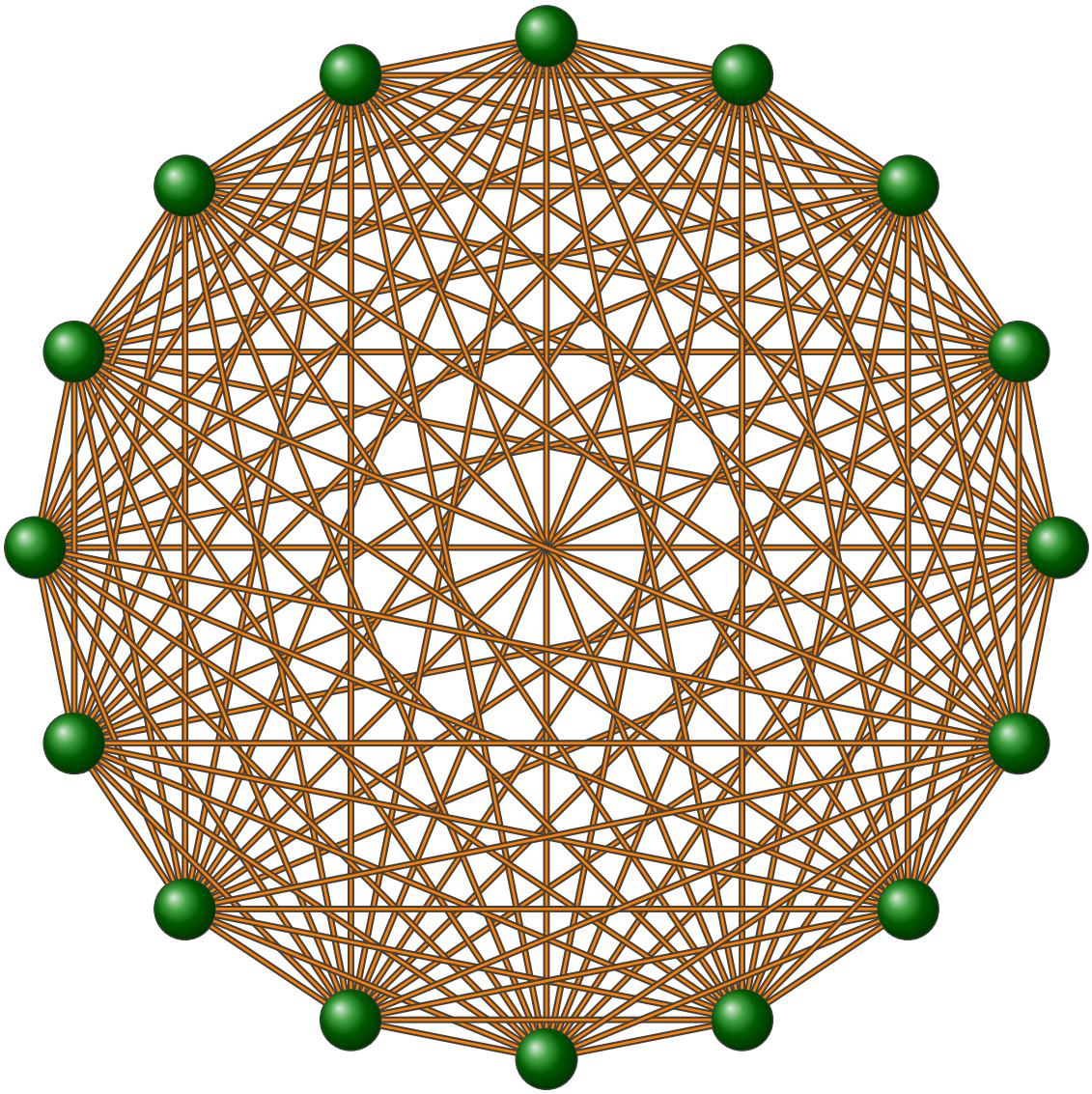
— Wikipedia
— MathWorld - Complete graph by E.Weisstein

3.0.4 Complete Graph order 4



```
\begin{tikzpicture}
  \renewcommand*{\VertexBallColor}{green!50!black}
  \GraphInit[vstyle=Shade]
  \grComplete[RA=5]{7}
\end{tikzpicture}
```

3.0.5 Complete Graph order 4



```
\begin{tikzpicture}
   \renewcommand*{\VertexBallColor}{green!50!black}
   \GraphInit[vstyle=Shade]
   \SetVertexNoLabel
   \grComplete[RA=7]{16}
\end{tikzpicture}
```

3.0.6 Circulant graph

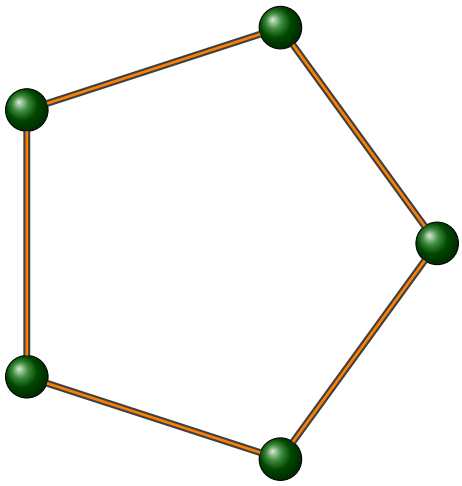
\grCirculant[⟨local options⟩]{⟨order⟩}

The circulant graph is defined for any order n at least 3, and every subset L of integers which are less than or equal to n/2. A circulant graph is a graph in which the i th graph vertex is adjacent to the (i + j)th and (i − j)th graph vertices for each j in a list L . The circulant graphs with L = {1; … ; [n/2]} gives the complete graphs and the circulant graph with L = {1} gives the cyclic graphs. The Möbius ladders are examples of circulant graphs.
In graph theory, a graph whose adjacency matrix is circulant is called a circulant graph.
The circulant graph on vertices on a list of nodes is implemented as **\grCirculant** *in the* **tkz-berge.sty** *package.*

External links :
MathWorld - CirculantGraph by E.Weisstein

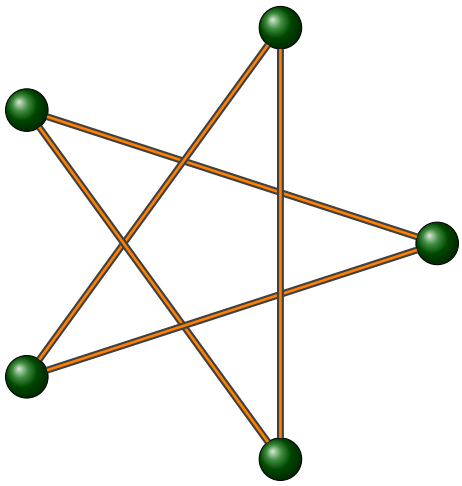
3.0.7 Graph order 5 with L={1}

This is a cycle graph.



```
\begin{tikzpicture}
   \grCirculant[RA=3]{5}{1}%
\end{tikzpicture}
```

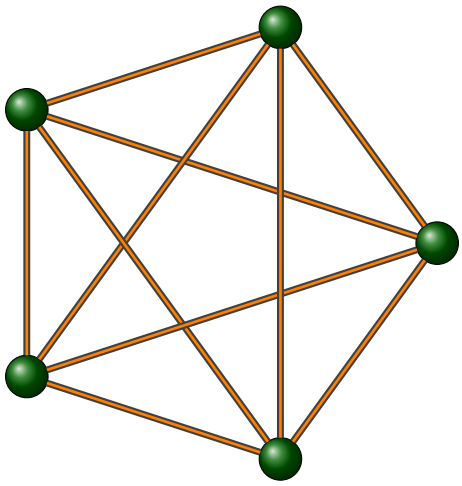
3.0.8 Graph order 5 with L={2}



```
\begin{tikzpicture}
   \grCirculant[RA=3]{5}{2}%
\end{tikzpicture}
```

3.0.9 Graph order 5 with L={1,2}

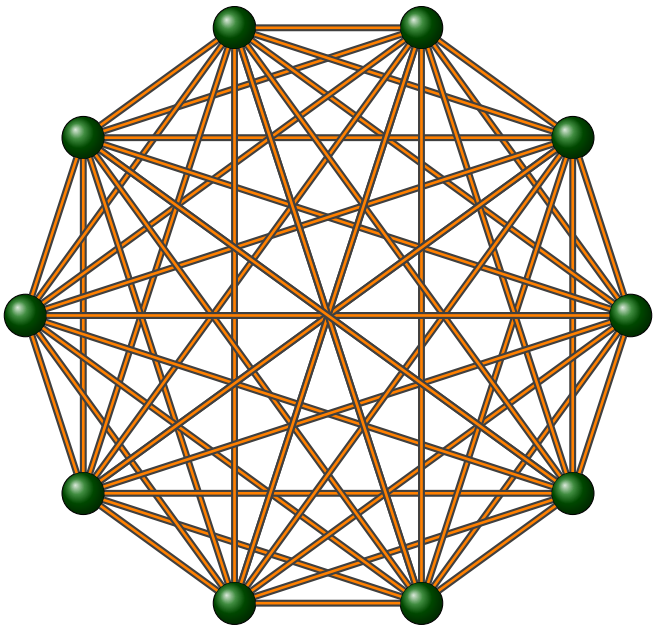
This graph is complete with an order 5.



```
\begin{tikzpicture}
   \grCirculant[RA=3]{5}{1,2}%
\end{tikzpicture}
```

3.0.10 Graph order 10 with L={1,2,3,4,5}

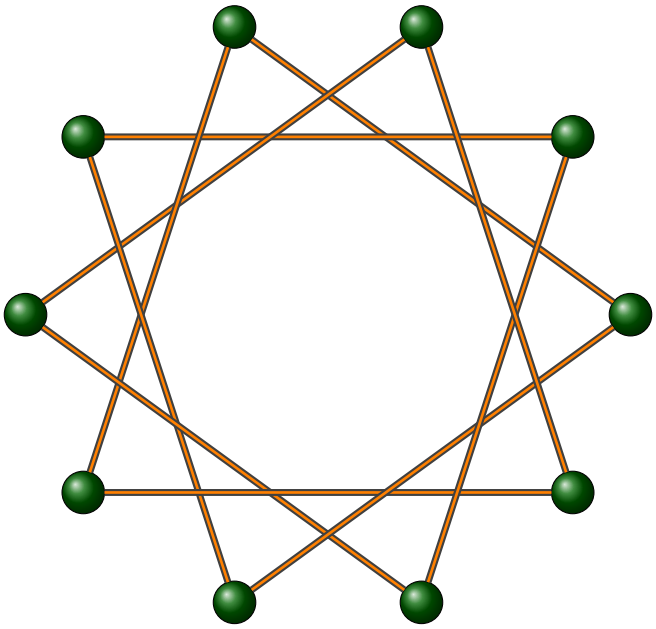
This graph is also complete



```
\begin{tikzpicture}
    \grCirculant[RA=4]{10}{1,2,3,4,5}%
\end{tikzpicture}
```

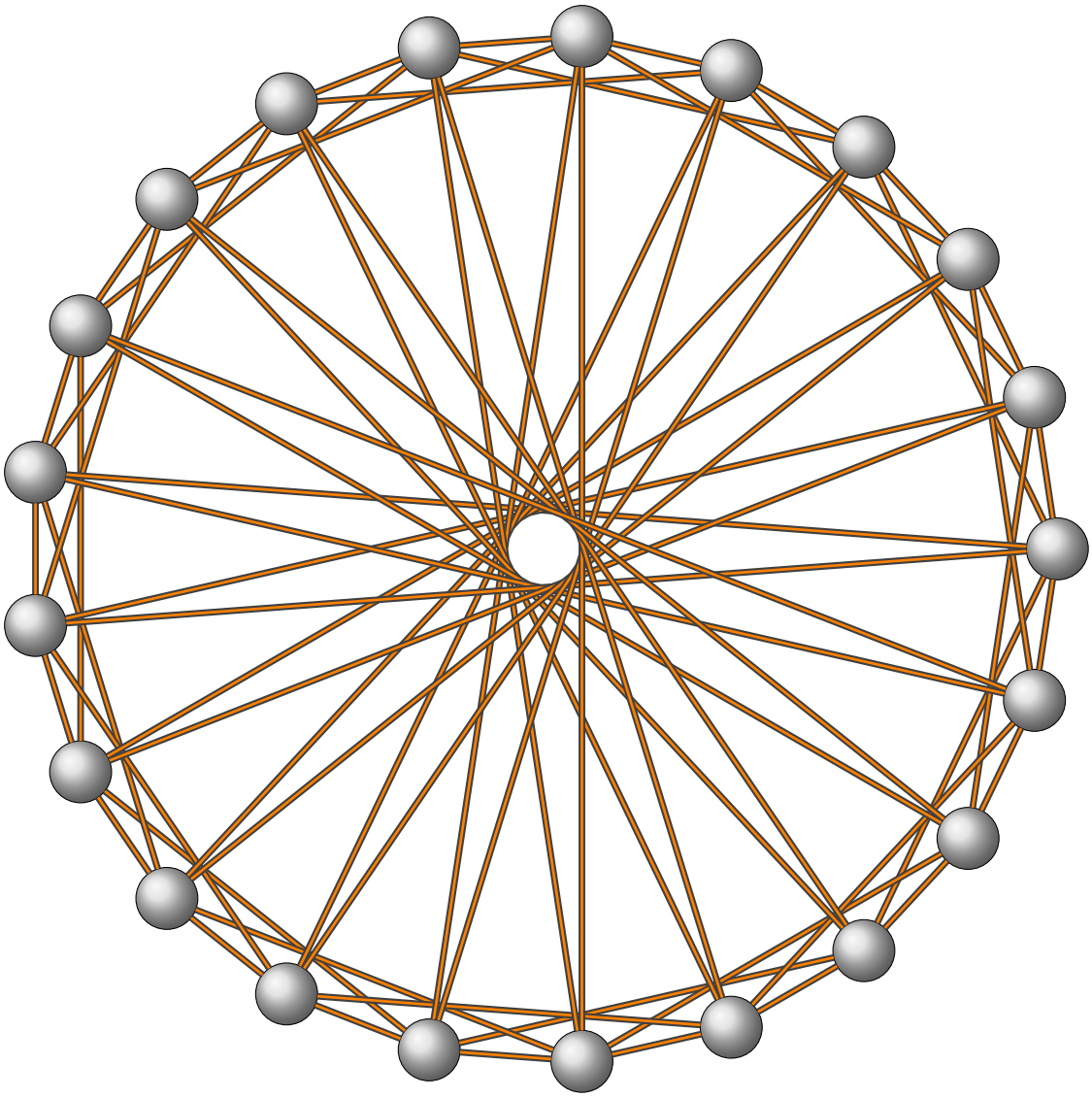
It's interesting to remark that the numbers 3 and 10 are primer, so if $L = \{3\}$ the graph is containing an Eulerian circuit.

3.0.11 Graph order 10 with L={3}



```
\begin{tikzpicture}
    \grCirculant[RA=4]{10}{3}%
\end{tikzpicture}
```

3.0.12 Graph order 21 with L={1,3,10}



3.0.13 Star graph

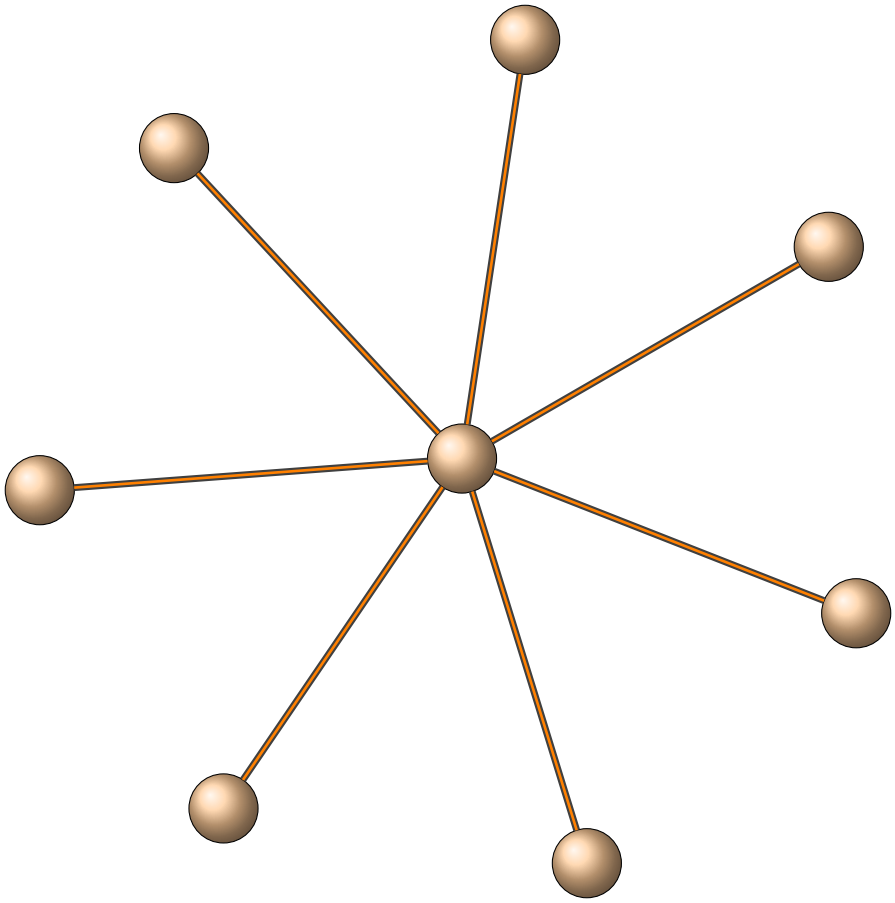
`\grStar[⟨local options⟩]{⟨order⟩}`

A star graph S_n is a n-graph with one node having vertex degree $n-1$ and the other $n-1$ having vertex degree 1. Star graphs can be generated using `\grStar` *in the* `tkz-berge.sty` *package.*

External links :

— MathWorld - StarGraph by Weisstein

3.0.14 Star graph



```
\begin{tikzpicture}[rotate=30,scale=.8]
  \grStar[RA=7]{8}%
\end{tikzpicture}
```

3.0.15 Square graph

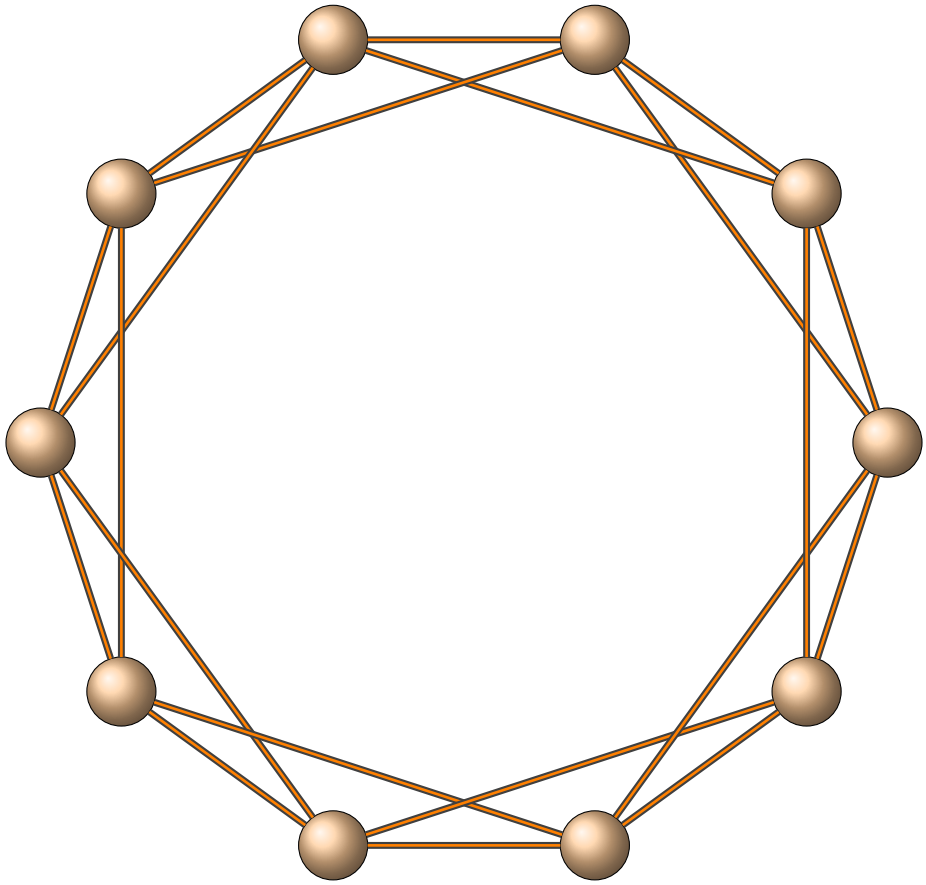
`\grSQCycle[⟨local options⟩]{⟨Number⟩}`

A star graph S_n is a n-graph with one node having vertex degree $n-1$ and the other $n-1$ having vertex degree 1. Star graphs can be generated using `\grStar` *in the* `tkz-berge.sty` *package.*

External links :

— MathWorld - SquareGraph by Weisstein

3.0.16 Square Cycle graph



```
\begin{tikzpicture}[scale=.8]
  \grSQCycle[RA=7]{10}%
\end{tikzpicture}
```

<———————————— WHEEL ————————————>

3.0.17 Wheel graph

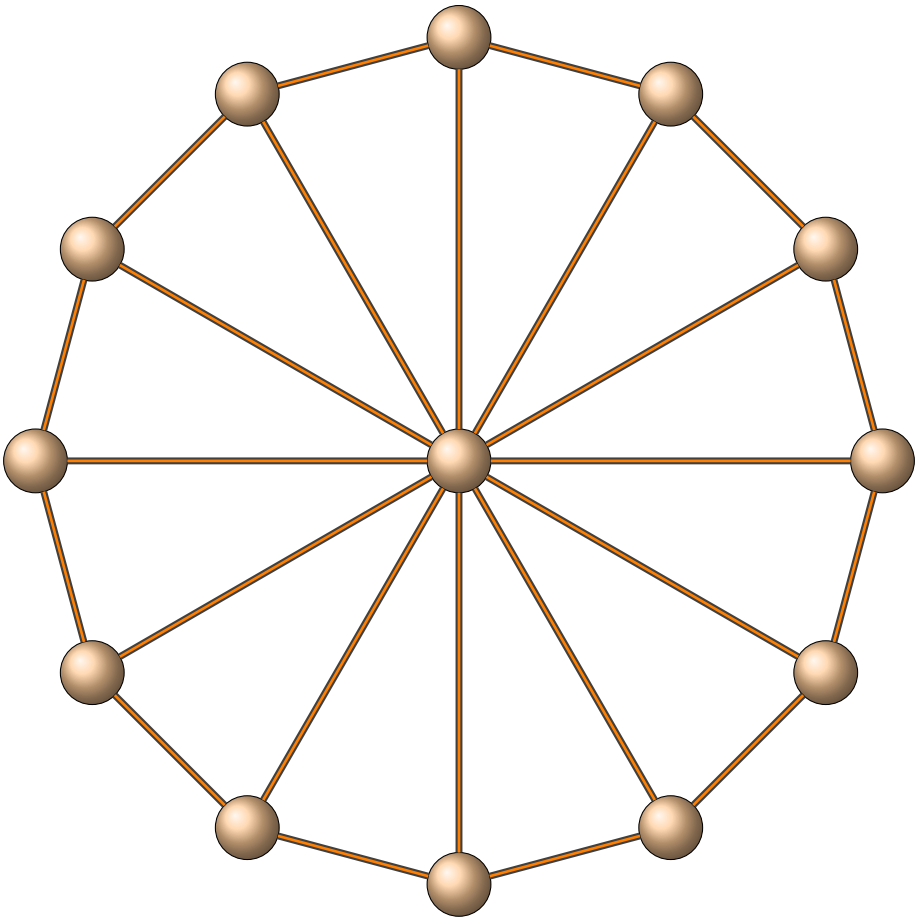
`\grWheel[⟨local options⟩]{⟨Number⟩}`

A wheel graph of order n is a graph that contains a cycle of order n − 1, and for which every vertex in the cycle is connected to one other vertex. The wheel can be defined as the graph , where is the singleton graph and is the cycle graph.

External links :

— MathWorld - WheelGraph by Weisstein

3.0.18 Wheel graph



```
\begin{tikzpicture}[scale=.8]
      \grWheel[RA=7]{13}%
\end{tikzpicture}
```

3.0.19 Ladder graph

`\grLadder[⟨local options⟩]{⟨Number⟩}`

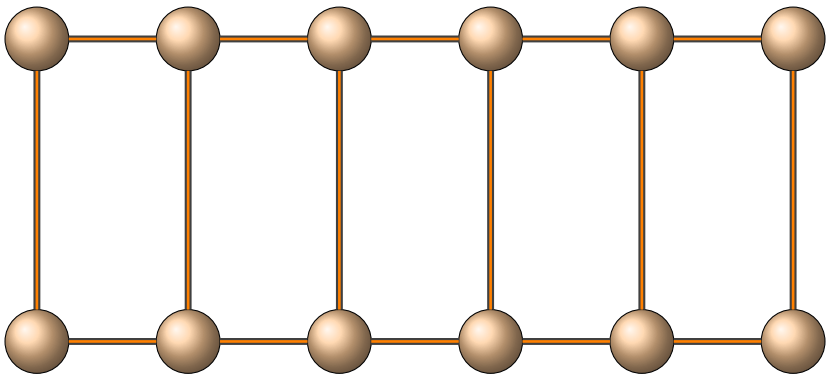
options	default	definition
RA	4	radius circle n°1
RS	0	distance between two lines
prefix	a	prefix for vertices
prefixx	b	prefix for vertices
Math	false	math mode

The ladder graph L_n or cyclic ladder graph is equivalent to the grid graph having two rails and n rungs between them.

External links :

— MathWorld - LadderGraph by Weisstein

3.0.20 Ladder graph



```
\begin{tikzpicture}
      \grLadder[RA=2,RS=4]{6}%
\end{tikzpicture}
```

3.0.21 Prism graph

`\grPrism[⟨local options⟩]{⟨Number⟩}`

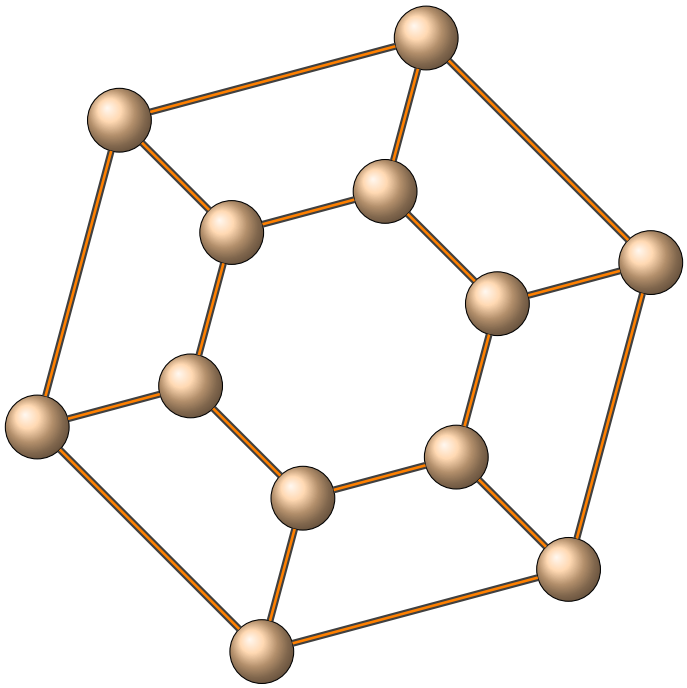
options	default	definition
RA	4	radius circle n°1
RB	3	radius circle n°2
prefix	a	prefix for vertices
prefixx	b	prefix for vertices
Math	false	math mode

An n-prism graph has $2n$ nodes and $3n$ edges, and is equivalent to the generalized Petersen graph with arguments n and 1. For odd n, the n-prism is isomorphic to the circulant graph with an order $2n$ and with arguments 2 and n. The 3-prism graph is the line graph of the complete bipartite graph with arguments 2 and 3 . The 4-prism graph is isomorphic with the cubical graph.

External links :

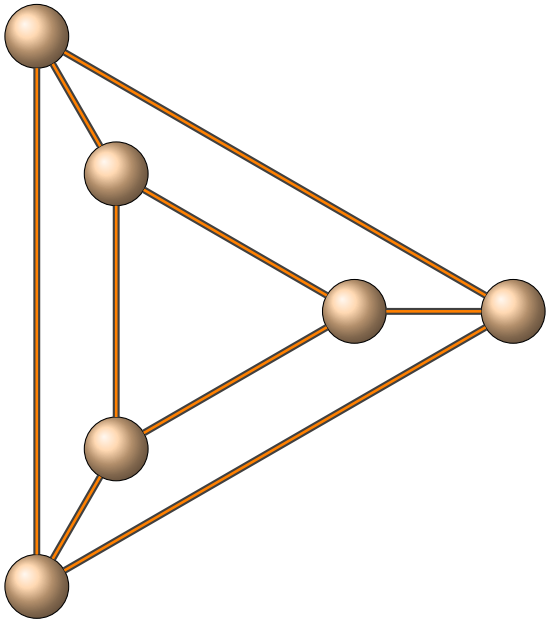
— MathWorld - Prism Graph by Weisstein

3.0.22 Cycle Ladder graph



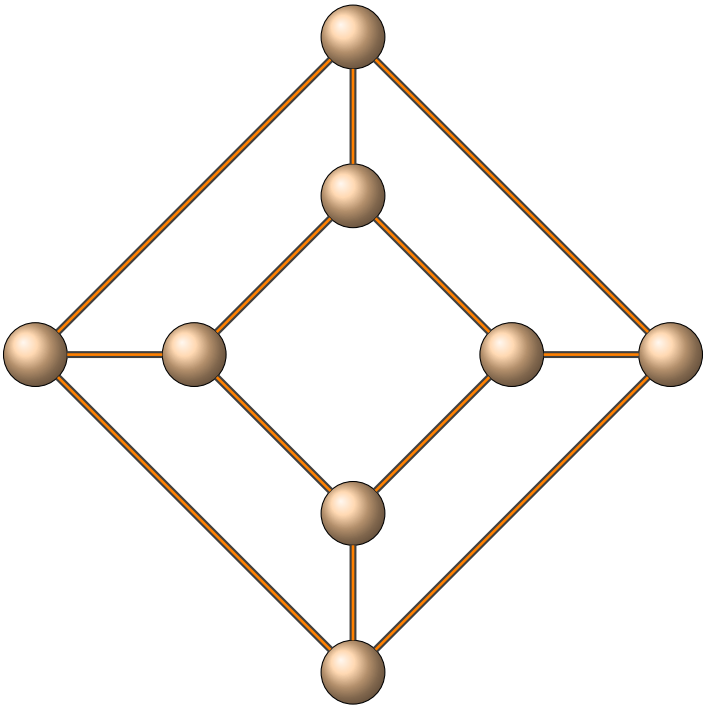
```
\begin{tikzpicture}[rotate=15,scale=.7]
  \grPrism[RA=6,RB=3]{6}%
\end{tikzpicture}
```

3.0.23 Cycle Ladder graph number 3



```
\begin{tikzpicture}[scale=.7]
  \grPrism[RA=6,RB=3]{3}%
\end{tikzpicture}
```

3.0.24 Cycle Ladder graph number 4



```
\begin{tikzpicture}[scale=.7]
  \grPrism[RA=6,RB=3]{4}%
\end{tikzpicture}
```

3.0.25 Complete Bipartite graph

`\grCompleteBipartite[⟨local options⟩]{⟨Number 1⟩}{⟨Number 2⟩}`

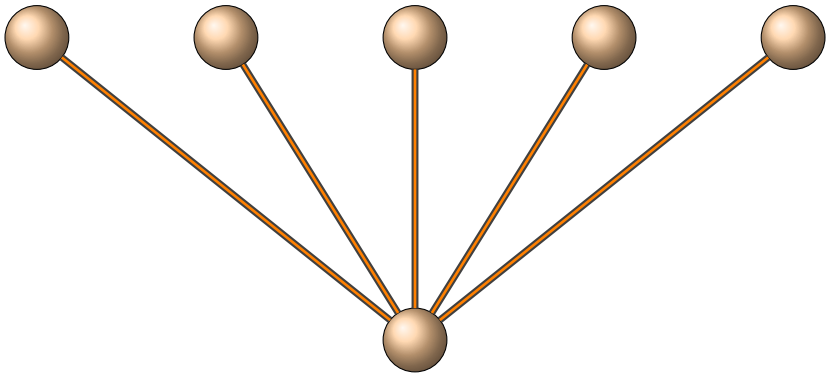
options	default	definition
RA	4	radius circle n°1
RB	3	radius circle n°2
RS	1	distance between two lines
form	1	integer to obtain a new embedding of a graph
prefix	a	prefix for vertices
prefixx	b	prefix for vertices
Math	false	math mode

A complete bipartite graph is a bipartite graph (i.e., a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent) such that every pair of graph vertices in the two sets are adjacent.

External links :

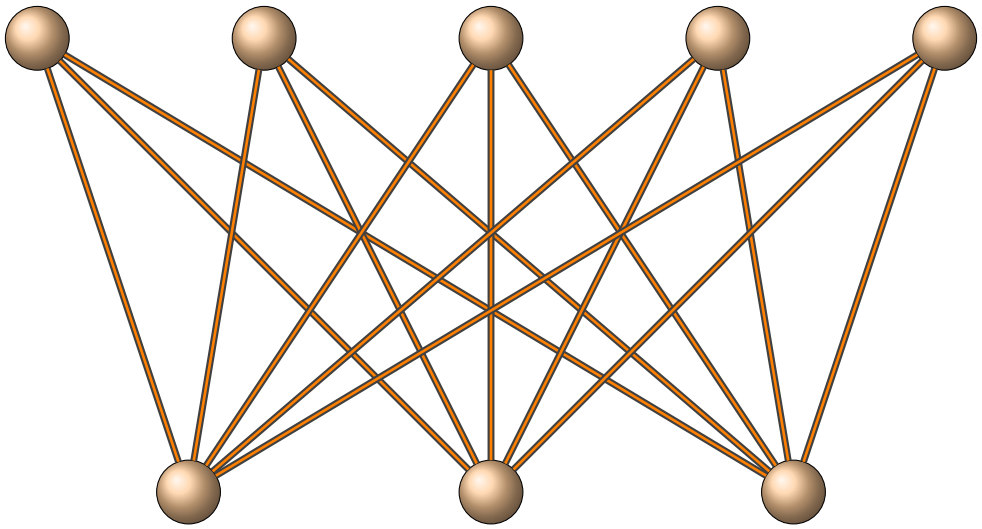
— MathWorld - CompleteBipartite Graph by Weisstein

3.0.26 Bipartite graph 1,5



```
\begin{tikzpicture}
   \grCompleteBipartite[RA=4,RB=2.5,RS=4]{1}{5}
\end{tikzpicture}
```

3.0.27 Bipartite graph 3,5



```
\begin{tikzpicture}
   \grCompleteBipartite[RA=4,RB=3,RS=6]{3}{5}
\end{tikzpicture}
```

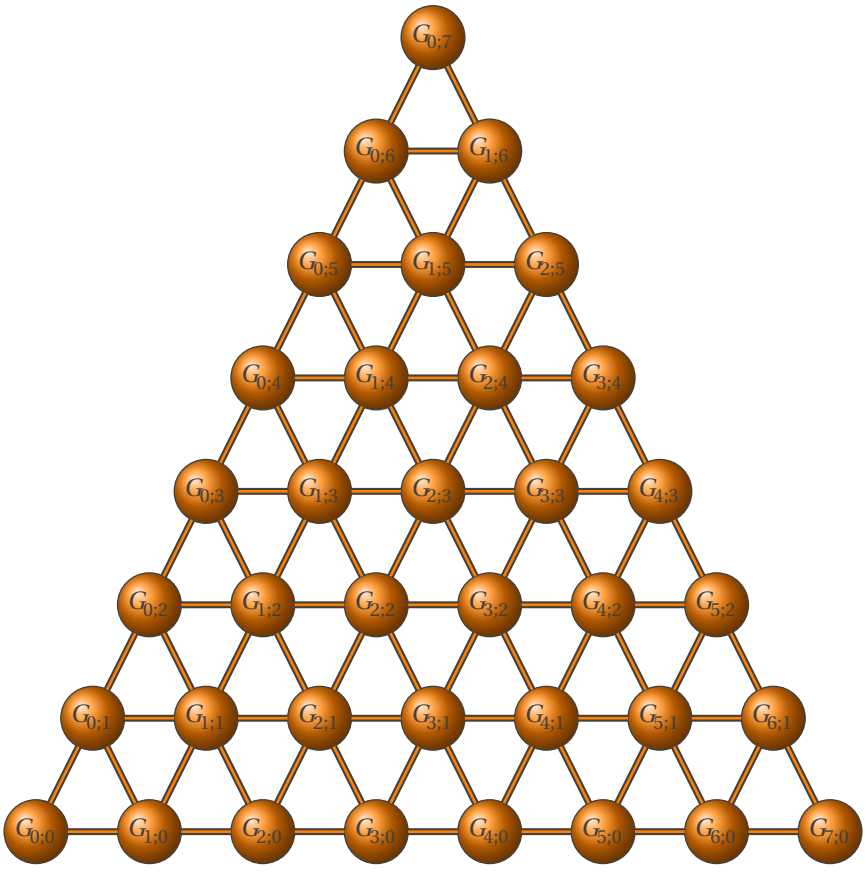
3.0.28 Triangular Grid graph

`\grTriangularGrid[⟨local options⟩]{⟨Number⟩}`

options	default	definition
RA	4	distance between two vertices
form	1	integer to obtain a new embedding of a graph
prefix	a	prefix for vertices
Math	false	math mode

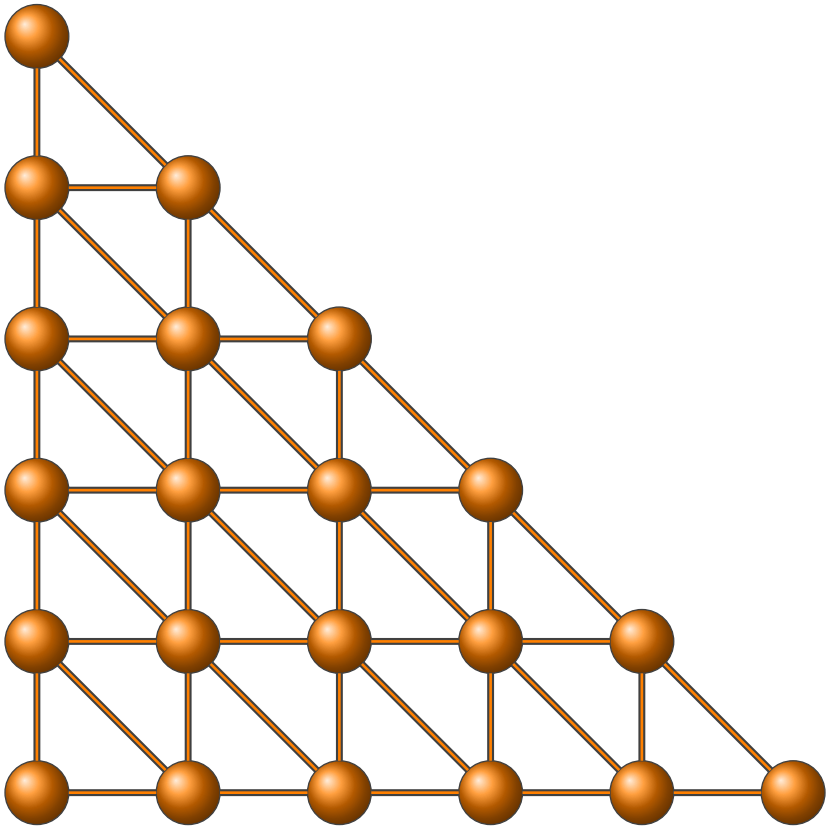
Number=n is the number of vertices of the first row then the graph order is $\frac{n(n-1)}{2}$. *There are three embeddings. You can use the option* **form** *with an integer between* 1 *and* 3.

3.0.29 n=8 order=28 form 1



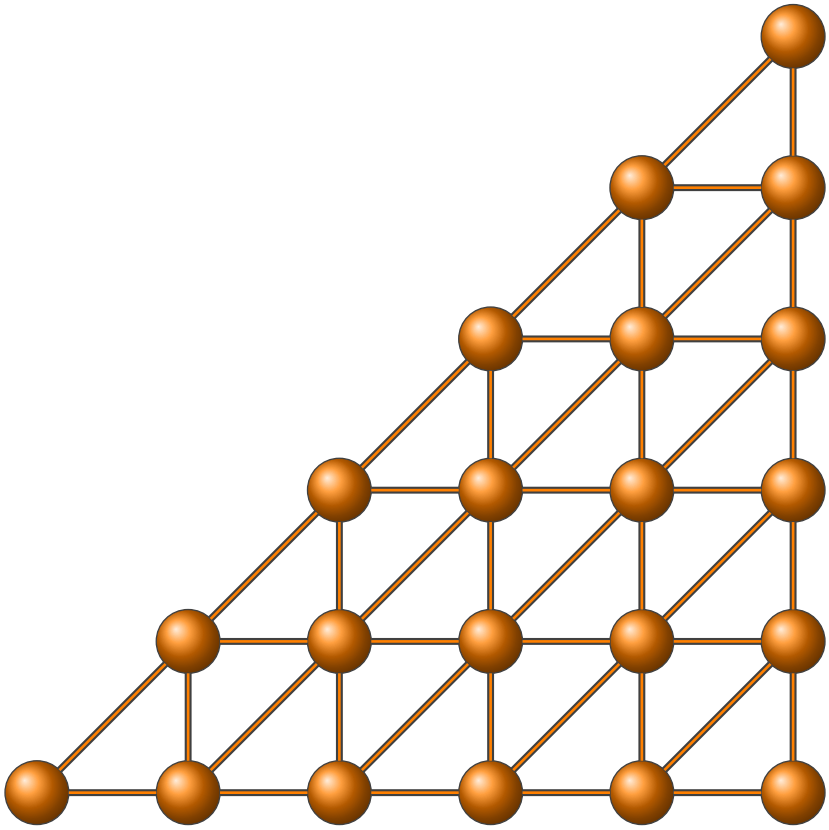

```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \SetVertexLabel
  \grTriangularGrid[prefix=G,Math,RA=1.5]{8}%
\end{tikzpicture}
```

3.0.30 n=6 order=15 form 2



```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \SetVertexNoLabel
  \grTriangularGrid[RA=2,form=2]{6}%
\end{tikzpicture}
```

3.0.31 n=6 order=15 form 3



```
\begin{tikzpicture}
  \GraphInit[vstyle=Shade]
  \SetVertexNoLabel
  \grTriangularGrid[RA=2,form=3]{6}%
\end{tikzpicture}
```

3.0.32 LCF Lederberg-Coxeter-Fruchte

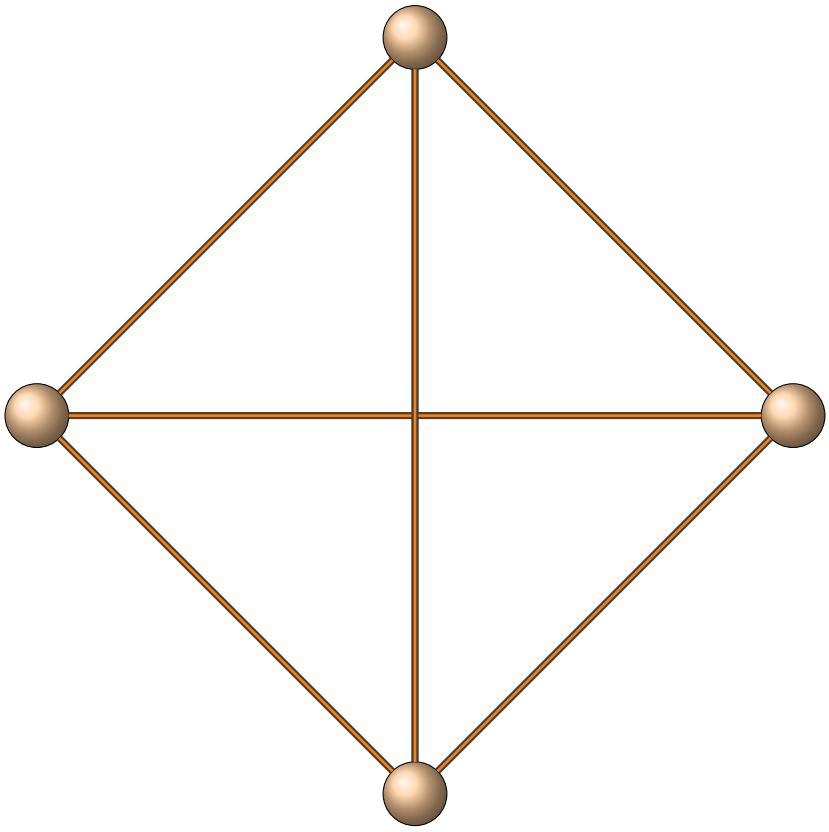
`\grLCF[⟨RA=⟨Number⟩⟩] {⟨List of numbers⟩} {⟨Number⟩}`

LCF = Lederberg-Coxeter-Fruchte (see the link below for some examples).

External links :

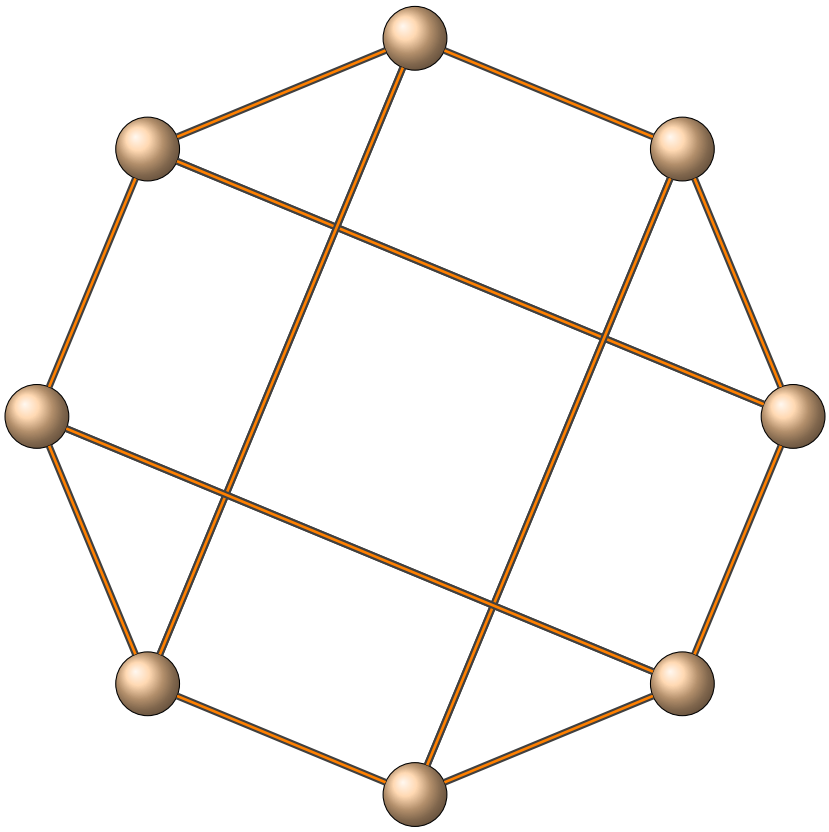
— MathWorld-LCF Notation by Weisstein

3.0.33 $[2,-2]^2$



```
\begin{tikzpicture}%
      \grLCF[RA=5]{2,-2}{2}%
\end{tikzpicture}
```

3.0.34 $[3,-3]^4$

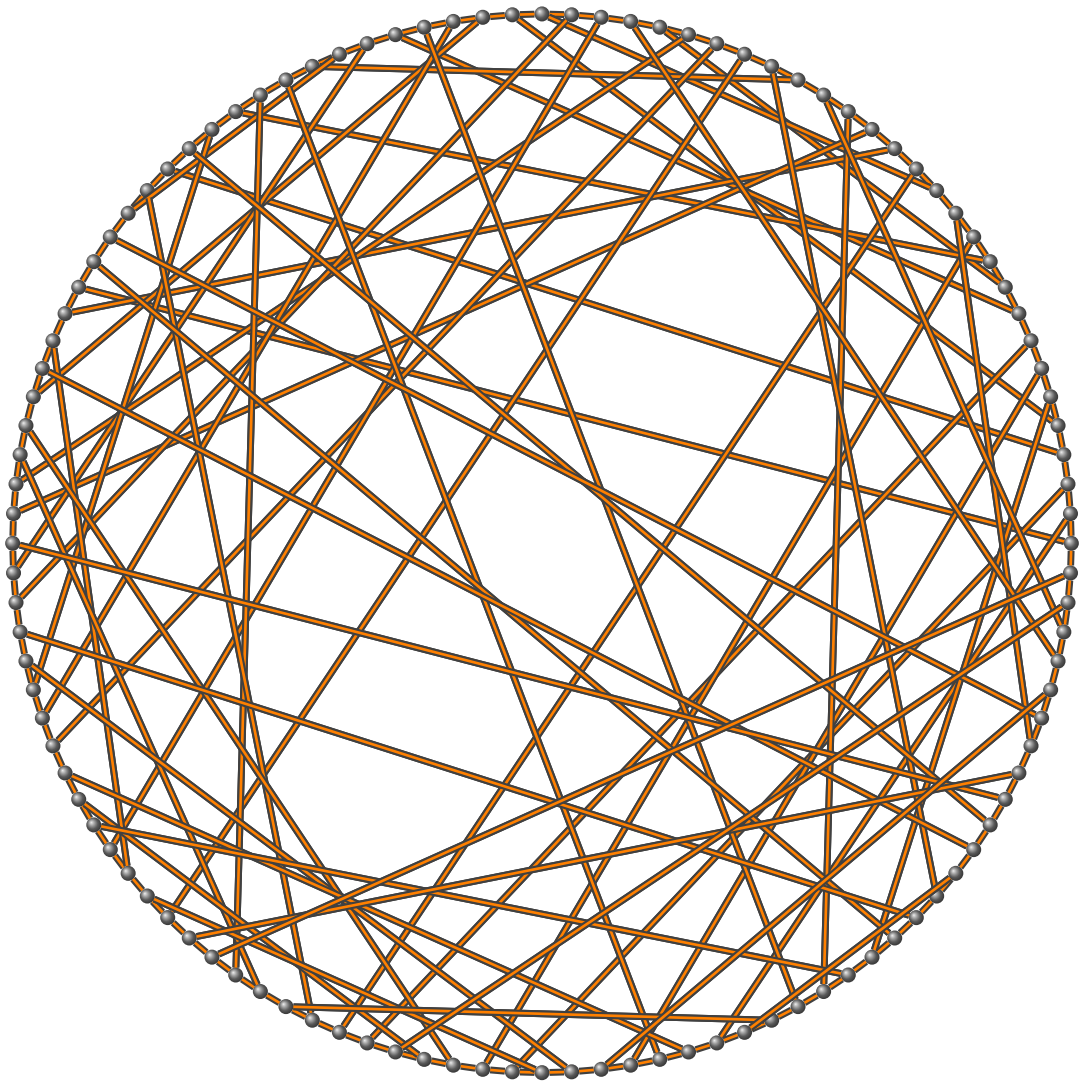


```
\begin{tikzpicture}%
       \grLCF[RA=5]{3,-3}{4}%
\end{tikzpicture}
```

3.0.35 Ljubljana graph

From Wikipedia http://en.wikipedia.org/wiki/Ljubljana_graph
The Ljubljana graph was first published in 1993 by Brouwer, Dejter and Thomassen. In 1972, Bouwer was already talking of a 112-vertices edge- but not vertex-transitive cubic graph found by R. M. Foster, but unpublished. Conder, Malnič, Marušič, Pisanski and Potočnik rediscovered this 112-vertices graph in 2002 and named it the Ljubljana graph after the capital of Slovenia. They proved that it was the unique 112-vertices edge- but not vertex-transitive cubic graph and therefore that was the graph found by Foster.
It can be represented in LCF notation as :

$$\Big[47,-23,-31,39,25,-21,-31,-41,25,15,29,-41,-19,15,-49,33,39,-35,-21,17,$$
$$-33,49,41,31,-15,-29,41,31,-15,-25,21,31,-51,-25,23,9,-17,51,35,-29,21,$$
$$-51,-39,33,-9,-51,51,-47,-33,19,51,-21,29,21,-31,-39\Big]^2$$



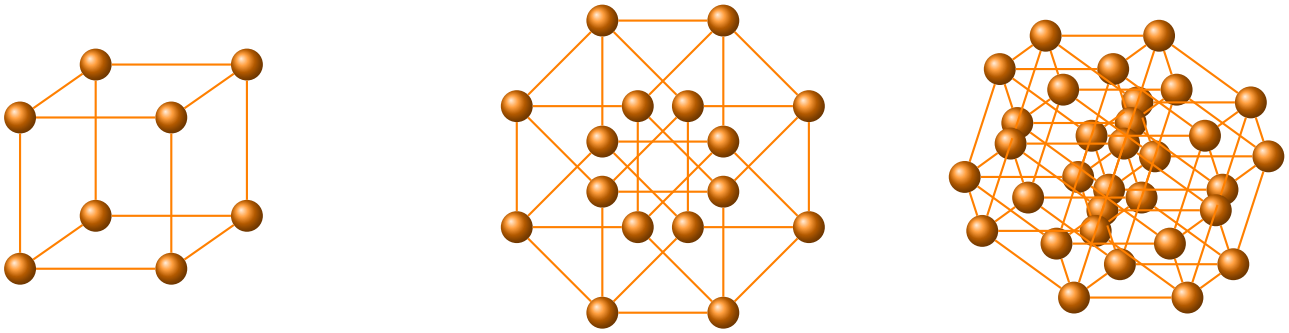
```
\GraphInit[vstyle=Art]
\SetGraphArtColor{black!50}{darkgray}
\tikzset{VertexStyle/.append style = {
                            minimum size         =  3pt}}
\begin{tikzpicture}%
 \grLCF[RA=7]{47, -23, -31, 39, 25, -21, -31, -41, 25, 15, 29, -41, -19, 15,%
  -49, 33, 39, -35, -21, 17, -33, 49, 41, 31, -15, -29, 41, 31, -15, -25, 21,%
   31, -51, -25, 23, 9, -17, 51, 35, -29, 21, -51, -39, 33, -9, -51, 51, -47,%
    -33, 19, 51, -21, 29, 21, -31, -39}{2}%
\end{tikzpicture}
```

4 New graphs and a gallery

4.1 Hypercubes: \grHypercube

\grHypercube[⟨local options⟩]{⟨n⟩}

The hypercube Q_n : 2^n vertices, the words of n bits, two words being adjacent when they differ by one bit. The vertex of the word $b_{n-1}\dots b_0$ is named **<prefix>**$b_{n-1}\dots b_0$ *(*a000, a001*…). The bit k moves a vertex along a vector of length* **RA**; *n = 1 to 4 have the classic drawings. Options :* **RA**, **prefix**, **Math**, **x**, **y**, **rotation**.

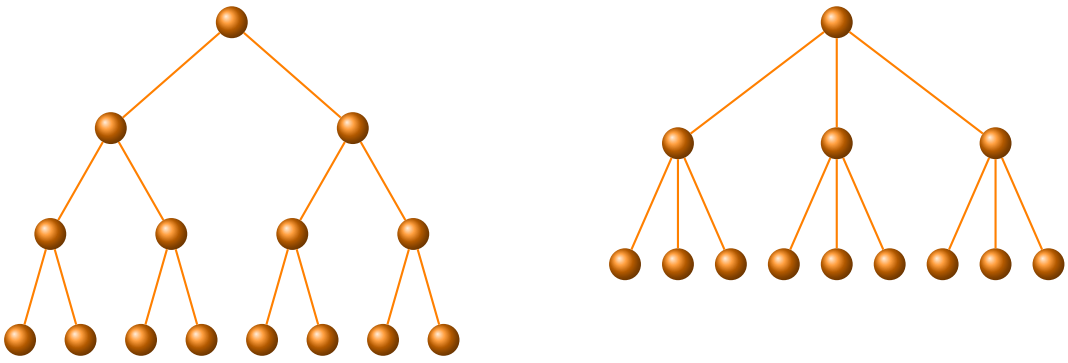


```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \grHypercube[RA=2]{3}
  \begin{scope}[xshift=7cm]
    \grHypercube[RA=1.6]{4}
  \end{scope}
  \begin{scope}[xshift=13cm]
    \grHypercube[RA=1.5,prefix=b]{5}
  \end{scope}
\end{tikzpicture}
```

4.2 Trees: \grTree

\grTree[⟨local options⟩]{⟨k⟩}{⟨depth⟩}

The complete k-ary tree; the root is at the top. The vertex j (j = 0,1 …) of the level l (the root is at the level 0) is named **<prefix>l;j**; *its parent is* **<prefix>(l-1);(j div k)**. *Options :* **RA** *(distance between two leaves),* **RB** *(distance between two levels),* **prefix**, **Math**, **x**, **y**.



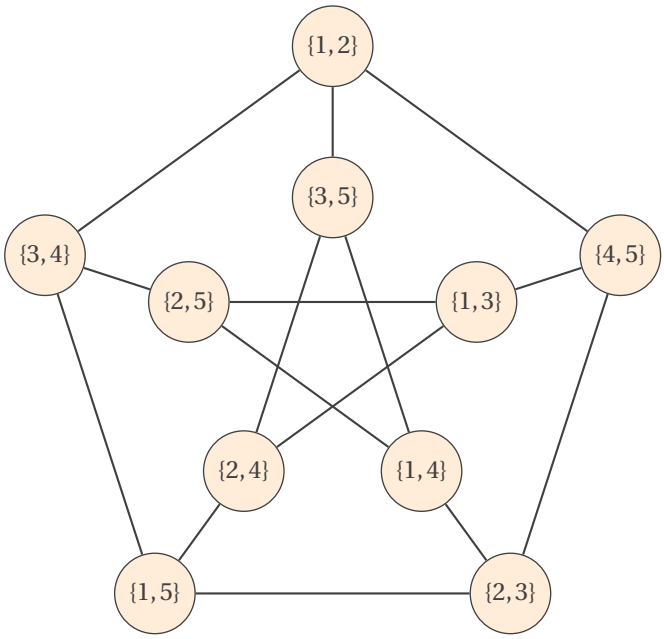
```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \grTree[RA=.8,RB=1.4]{2}{3}
  \begin{scope}[xshift=8cm]
    \grTree[RA=.7,RB=1.6]{3}{2}
  \end{scope}
\end{tikzpicture}
```

4.3 A gallery

Some graphs of graph theory, drawn with the macros of **tkz-graph** and **tkz-berge**. The complete files are in the folder **examples** of the package.

4.3.1 The Petersen graph is a Kneser graph

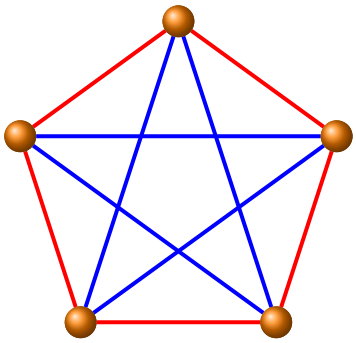
The vertices of the Kneser graph $K(n,k)$ are the subsets of k elements of $\{1,\dots,n\}$; two subsets are adjacent when they are disjoint. $K(5,2)$ is the Petersen graph. The edges are not given : they are computed with `\pgfmathtruncatemacro`, the test `&&` giving 1 for two disjoint subsets.



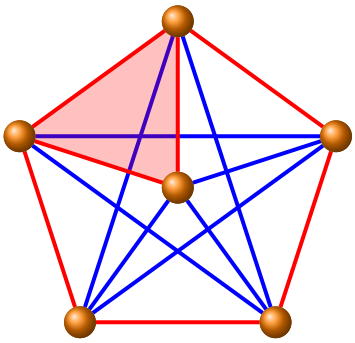

```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \tikzset{VertexStyle/.append style = {minimum size = 26pt,
                                        fill = orange!15}}
  \SetVertexMath
  \foreach \a/\b/\k in {1/2/0,3/4/1,1/5/2,2/3/3,4/5/4}
    {\Vertex[a=90+72*\k,d=4,L={\{\a,\b\}}]{\a\b}}
  \foreach \a/\b/\k in {3/5/0,2/5/1,2/4/2,1/4/3,1/3/4}
    {\Vertex[a=90+72*\k,d=2,L={\{\a,\b\}}]{\a\b}}
  \foreach \a/\b in {1/2,1/3,1/4,1/5,2/3,2/4,2/5,3/4,3/5,4/5}{%
    \foreach \c/\d in {1/2,1/3,1/4,1/5,2/3,2/4,2/5,3/4,3/5,4/5}{%
      \pgfmathtruncatemacro\ok{(\a!=\c)&&(\a!=\d)&&(\b!=\c)&&(\b!=\d)
                               &&(\a\b<\c\d)}
      \ifnum\ok=1 \Edge(\a\b)(\c\d)\fi}}
\end{tikzpicture}
```

4.3.2 Ramsey: $R(3,3) = 6$

Two colors for the edges of K_5 are enough to avoid a monochromatic triangle (a red pentagon and a blue pentagram), but not for K_6 : one of the five edges from f has the same color as two others, and the triangle is red or blue. The Ramsey number $R(3,3)$ is 6.



K_5 : no monochromatic triangle

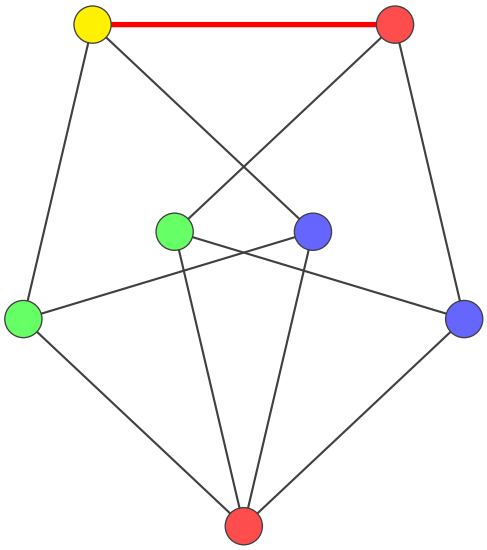


K_6 : the triangle fab is red

```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \SetGraphUnit{2.2}
  \Vertices[rotation=90]{circle}{0,1,2,3,4}
  \Edges[local,color=red,lw=1.5pt](0,1,2,3,4,0)
  \Edges[local,color=blue,lw=1.5pt](0,2,4,1,3,0)
  \Vertices[rotation=90]{circle}{0,1,2,3,4}
  \node at (0,-3) {\(K_5\) : no monochromatic triangle};
  \begin{scope}[xshift=7cm]
    \Vertices[rotation=90]{circle}{a,b,c,d,e}
    \Vertex[x=0,y=0]{f}
    \Edges[local,color=red,lw=1.5pt](a,b,c,d,e,a)
    \Edges[local,color=blue,lw=1.5pt](a,c,e,b,d,a)
    \foreach \v in {a,b}{\Edge[local,color=red,lw=1.5pt](f)(\v)}
    \foreach \v in {c,d,e}{\Edge[local,color=blue,lw=1.5pt](f)(\v)}
    \fill[red,opacity=.25] (f.center) -- (a.center) -- (b.center) -- cycle;
    \Vertices[rotation=90]{circle}{a,b,c,d,e}
    \Vertex[x=0,y=0]{f}
    \node at (0,-3) {\(K_6\) : the triangle \(f a b\) is red};
  \end{scope}
\end{tikzpicture}
```

4.3.3 The Moser spindle

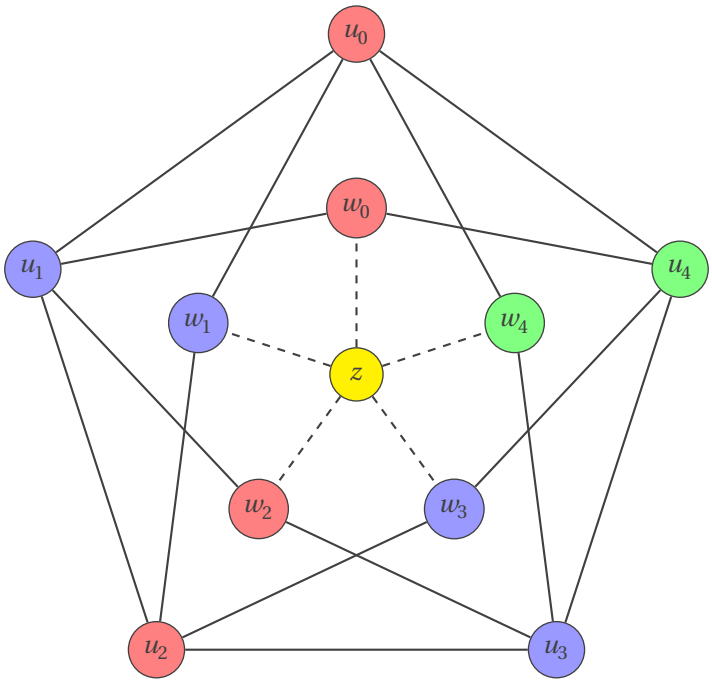
Seven points and eleven segments of length 1; four colors are necessary, so at least four colors are necessary to color the points of the plane when two points at distance 1 have different colors (Hadwiger–Nelson problem). The two spindles are turned by the angle $\theta = 2\arcsin\frac{1}{2\sqrt{3}}$ so that their tips are at distance 1.



```
\begin{tikzpicture}[scale=4]
  \GraphInit[vstyle=Normal]
  \SetVertexNoLabel
  \tikzset{VertexStyle/.append style = {minimum size = 14pt}}
  \pgfmathsetmacro\t{2*asin(1/(2*sqrt(3)))}
  \pgfmathsetmacro\u{90-\t/2}
  \pgfmathsetmacro\w{90+\t/2}
  \tikzset{c1/.style={fill=red!70},c2/.style={fill=blue!60},
           c3/.style={fill=green!60},c4/.style={fill=yellow}}
  \coordinate (A) at (0,0);
  \coordinate (B) at (\u-
30:1);  \coordinate (C) at (\u+30:1);
  \coordinate (D) at (\u:{sqrt(3)});
  \coordinate (E) at (\w-
30:1);  \coordinate (F) at (\w+30:1);
  \coordinate (G) at (\w:{sqrt(3)});
  \Edges(A,B,D,C,A) \Edge(B)(C)
  \Edges(A,E,G,F,A) \Edge(E)(F)
  \Edge[local,color=red,lw=2pt](D)(G)
  \foreach \v/\c in {A/c1,B/c2,C/c3,D/c1,E/c2,F/c3,G/c4}
    {\node[VertexStyle,\c] at (\v) {};}
\end{tikzpicture}
```

4.3.4 Mycielski's construction

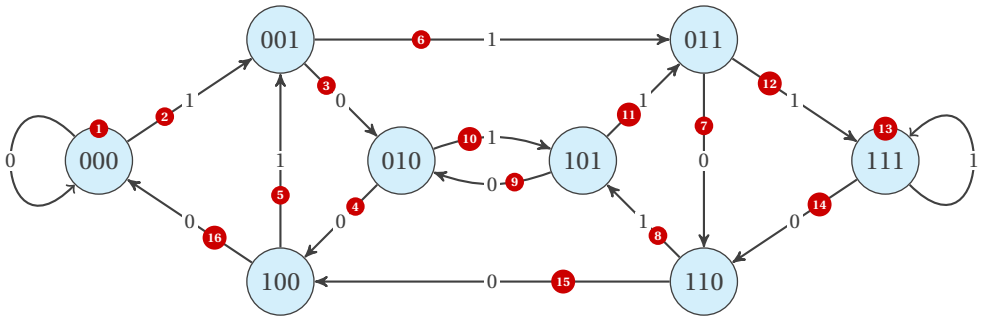
From a graph without triangle, Mycielski's construction gives a graph without triangle that needs one more color. Each vertex u_i of C_5 receives a shadow w_i joined to the neighbors of u_i, and a vertex z is joined to all the shadows : the result is the Grötzsch graph (`\grGrotzsch`), with 4 colors.




```
\begin{tikzpicture}
  \GraphInit[vstyle=Normal]
  \SetVertexMath
  \tikzset{VertexStyle/.append style = {minimum size = 20pt}}
  \foreach \i/\c in {0/red!50,1/blue!40,2/red!50,3/blue!40,4/green!50}{%
    \begin{scope}[VertexStyle/.append style = {fill = \c}]
      \Vertex[a=90+72*\i,d=4.5,L={u_{\i}}]{u\i}
      \Vertex[a=90+72*\i,d=2.2,L={w_{\i}}]{w\i}
    \end{scope}}
  \begin{scope}[VertexStyle/.append style = {fill = yellow}]
    \Vertex[x=0,y=0,L=z]{z}
  \end{scope}
  \Edges[lw=1.5pt](u0,u1,u2,u3,u4,u0)
  \foreach \i in {0,...,4}{%
    \pgfmathtruncatemacro\p{mod(\i+4,5)}
    \pgfmathtruncatemacro\n{mod(\i+1,5)}
    \Edge[color=gray](w\i)(u\p)
    \Edge[color=gray](w\i)(u\n)
    \Edge[color=orange,style={dashed}](z)(w\i)}
\end{tikzpicture}
```

4.3.5 The de Bruijn graph $B(2,3)$

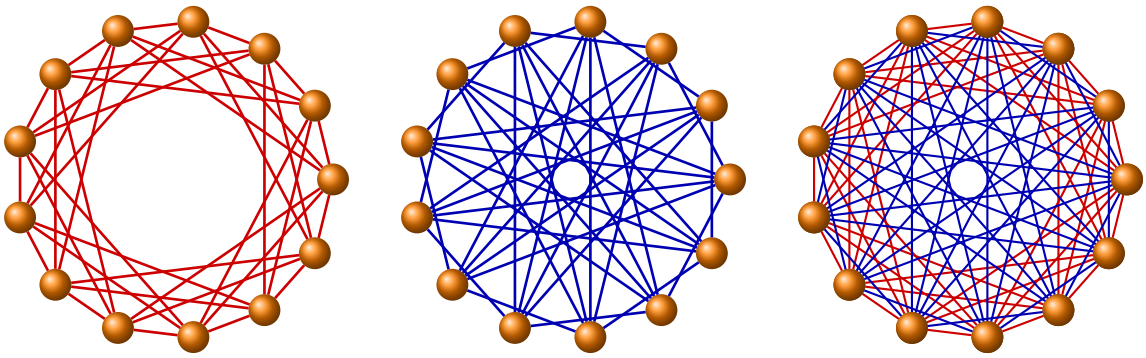
The vertices are the words of three bits, and there is an arc $abc \to bcd$ labelled d. Each vertex has two arcs in and two arcs out : there is an Eulerian cycle. The bits read in the order of the numbers give a de Bruijn sequence : each word of four bits appears once in it. This example needs `lualatex` and the option `lua` (`\tkzGraphEuler`).




```
\begin{tikzpicture}[scale=.8]
  \GraphInit[vstyle=Normal]
  \tikzset{VertexStyle/.append style = {minimum size = 24pt,
                                        fill = cyan!15}}
  \tikzset{LabelStyle/.style = {fill = white, font = \footnotesize,
                                inner sep = 1pt}}
  \tikzset{EulerNumberStyle/.style = {circle, fill = red!80!black,
           text = white, inner sep = 1pt, font = \tiny\bfseries}}
  \Vertex[x=0,y=0]{000}
  \Vertex[x=3,y=2]{001}    \Vertex[x=3,y=-2]{100}
  \Vertex[x=5,y=0]{010}    \Vertex[x=8,y=0]{101}
  \Vertex[x=10,y=2]{011}   \Vertex[x=10,y=-2]{110}
  \Vertex[x=13,y=0]{111}
  \Loop[dir=WE,dist=2cm,style={->},label=0](000)
  \Loop[dir=EA,dist=2cm,style={->},label=1](111)
  \Edge[->,label=1](000)(001)
  \Edge[->,label=0](001)(010)
  \Edge[->,label=1](001)(011)
  \Edge[->,label=0](010)(100)
  \Edge[->,label=1,style={bend left=20}](010)(101)
  \Edge[->,label=0,style={bend left=20}](101)(010)
  \Edge[->,label=1](101)(011)
  \Edge[->,label=0](011)(110)
  \Edge[->,label=1](011)(111)
  \Edge[->,label=0](100)(000)
  \Edge[->,label=1](100)(001)
  \Edge[->,label=0](110)(100)
  \Edge[->,label=1](110)(101)
  \Edge[->,label=0](111)(110)
  \tkzGraphEuler[numbers,pos=.3,start=000]
\end{tikzpicture}
```

4.3.6 The Paley graph P_{13}

In P_{13}, i and j are adjacent when $i-j$ is a square modulo 13 (±1, ±3, ±4) : it is the circulant graph `\grCirculant` with $1,3,4$. Its complement $(2,5,6)$ is isomorphic to it : the edges of K_{13} are split into two copies of the same graph.

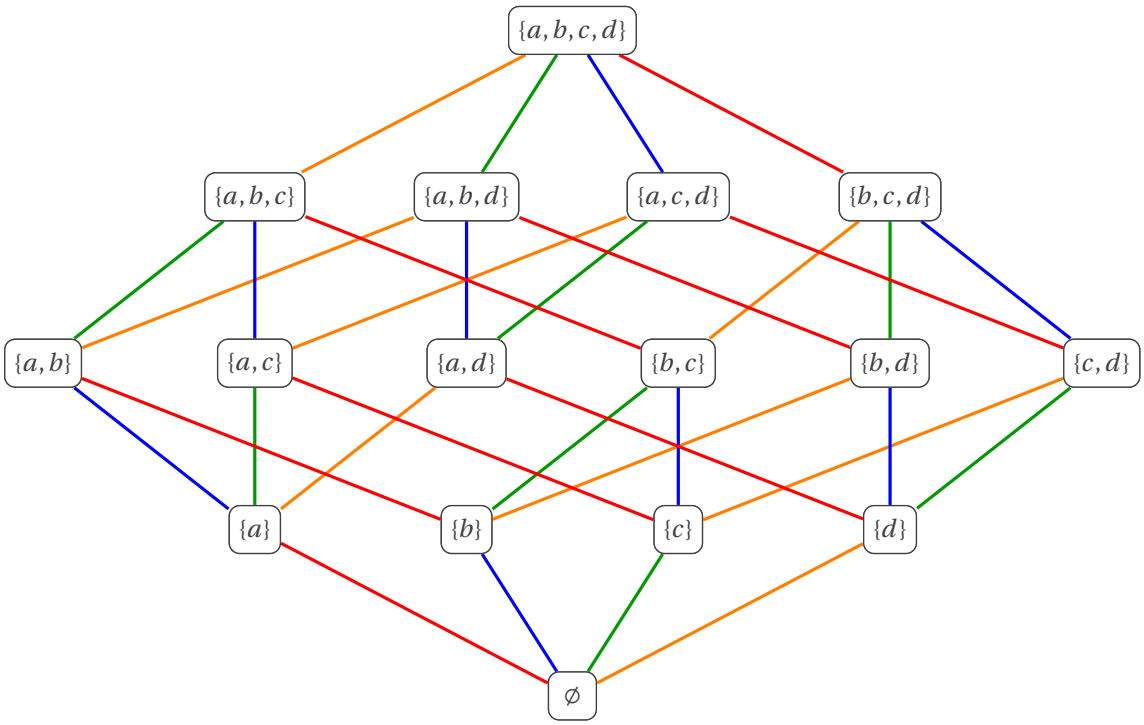


P_{13} $\overline{P_{13}} \simeq P_{13}$ $K_{13} = P_{13} \cup \overline{P_{13}}$

```
\begin{tikzpicture}[scale=.7]
  \GraphInit[vstyle=Art]
  \SetUpEdge[color=red!80!black,lw=1pt]
  \grCirculant[RA=3]{13}{1,3,4}
  \begin{scope}[xshift=7.5cm]
    \SetUpEdge[color=blue!70!black,lw=1pt]
    \grCirculant[RA=3]{13}{2,5,6}
  \end{scope}
  \begin{scope}[xshift=15cm]
    \SetUpEdge[color=red!80!black,lw=.8pt]
    \grCirculant[RA=3]{13}{1,3,4}
    \SetUpEdge[color=blue!70!black,lw=.8pt]
    \grCirculant[RA=3]{13}{2,5,6}
  \end{scope}
  \node at (0,-4)    {\(P_{13}\)};
  \node at (7.5,-4)  {\(\overline{P_{13}}\simeq P_{13}\)};
  \node at (15,-4)   {\(K_{13}=P_{13}\cup\overline{P_{13}}\)};
\end{tikzpicture}
```

4.3.7 The hypercube Q_4 and the subsets of a set

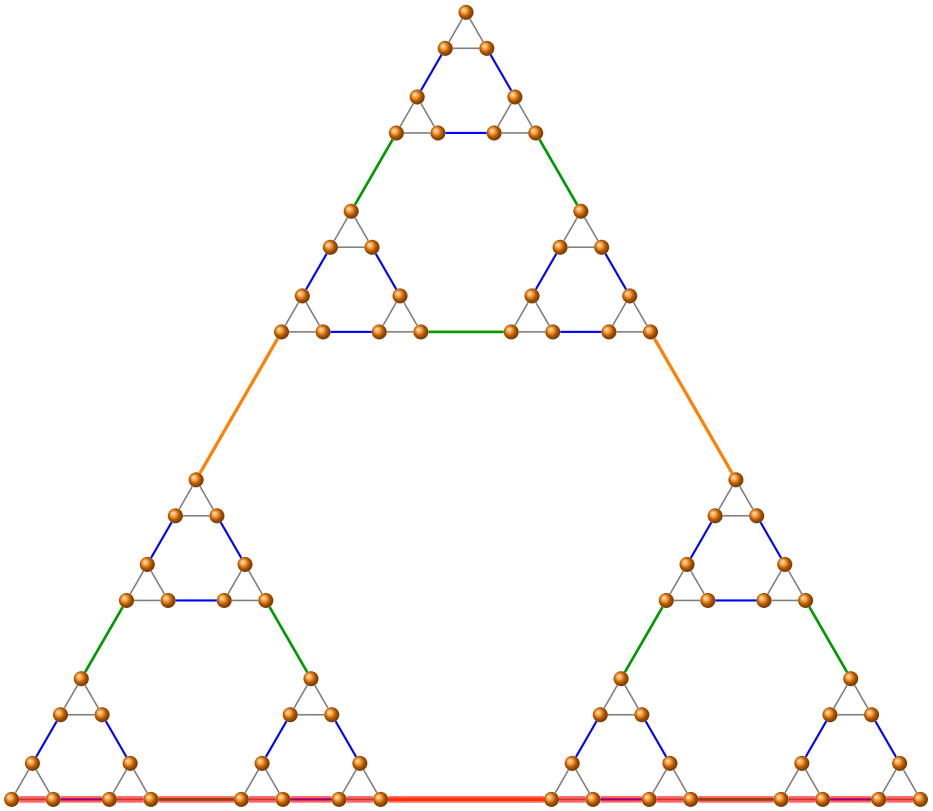
Q_4 drawn as the Hasse diagram of the subsets of $\{a,b,c,d\}$: an edge joins two subsets when one is obtained by adding an element to the other; its color gives this element. Compare with `\grHypercube`.



```
\begin{tikzpicture}[xscale=1.4,yscale=2.2]
  \GraphInit[vstyle=Normal]
  \SetVertexMath
  \tikzset{VertexStyle/.append style = {rectangle, rounded corners,
            minimum size = 18pt, fill = white}}
  \foreach \n/\x/\y/\l in {
     0/0/0/{\emptyset},
     a/-3/1/{\{a\}}, b/-1/1/{\{b\}}, c/1/1/{\{c\}}, d/3/1/{\{d\}},
     ab/-5/2/{\{a,b\}}, ac/-3/2/{\{a,c\}}, ad/-1/2/{\{a,d\}},
     bc/1/2/{\{b,c\}},  bd/3/2/{\{b,d\}},  cd/5/2/{\{c,d\}},
     abc/-3/3/{\{a,b,c\}}, abd/-1/3/{\{a,b,d\}},
     acd/1/3/{\{a,c,d\}},  bcd/3/3/{\{b,c,d\}},
     abcd/0/4/{\{a,b,c,d\}}}
   {\Vertex[x=\x,y=\y,L=\l]{\n}}
  \foreach \s/\t/\c in {
    0/a/red, 0/b/blue, 0/c/green!60!black, 0/d/orange, a/ab/blue,
    a/ac/green!60!black, a/ad/orange, b/ab/red, b/bc/green!60!black,
    b/bd/orange, c/ac/red, c/bc/blue, c/cd/orange, d/ad/red,
    d/bd/blue, d/cd/green!60!black, ab/abc/green!60!black,
    ab/abd/orange, ac/abc/blue, ac/acd/orange, ad/abd/blue,
    ad/acd/green!60!black, bc/abc/red, bc/bcd/orange, bd/abd/red,
    bd/bcd/green!60!black, cd/acd/red, cd/bcd/blue, abc/abcd/orange,
    abd/abcd/green!60!black, acd/abcd/blue, bcd/abcd/red}
    {\Edge[local,color=\c,lw=1.2pt](\s)(\t)}
  \foreach \n/\x/\y/\l in {
     0/0/0/{\emptyset},
     a/-3/1/{\{a\}}, b/-1/1/{\{b\}}, c/1/1/{\{c\}}, d/3/1/{\{d\}},
     ab/-5/2/{\{a,b\}}, ac/-3/2/{\{a,c\}}, ad/-1/2/{\{a,d\}},
     bc/1/2/{\{b,c\}},  bd/3/2/{\{b,d\}},  cd/5/2/{\{c,d\}},
     abc/-3/3/{\{a,b,c\}}, abd/-1/3/{\{a,b,d\}},
     acd/1/3/{\{a,c,d\}},  bcd/3/3/{\{b,c,d\}},
     abcd/0/4/{\{a,b,c,d\}}}
   {\Vertex[x=\x,y=\y,L=\l]{\n}}
\end{tikzpicture}
```

4.3.8 The Tower of Hanoi

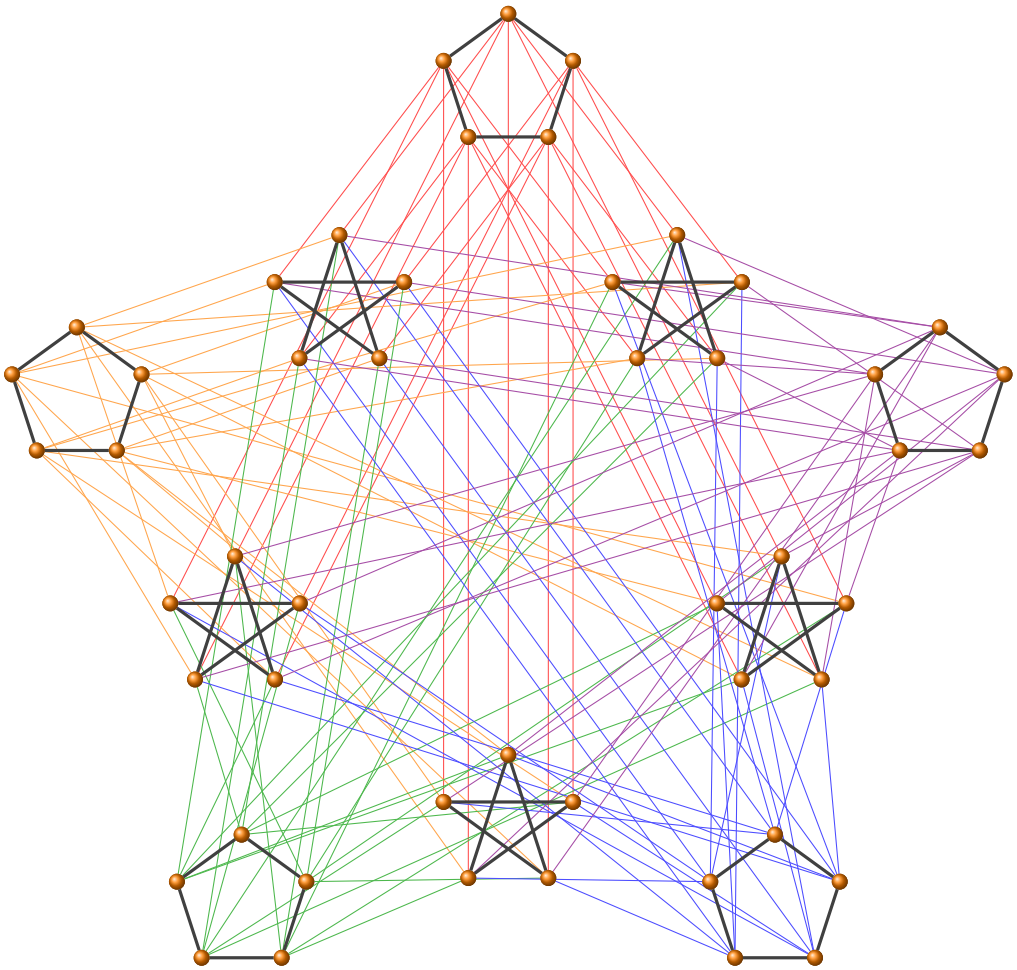
The 81 states of the Tower of Hanoi with 4 discs; two states are adjacent when one move goes from one to the other. This graph is the Sierpiński graph $S(4,3)$: a vertex $d_1d_2d_3d_4$ is placed at the sum of the corners $c(d_k)$ multiplied by L^{4-k}, and $s\,i\,j\ldots j$ is adjacent to $s\,j\,i\ldots i$. The colors give the size of the disc moved. The bottom side, from 0000 to 2222, is an optimal solution (15 moves) : the words of $\{0,2\}^4$ in the order of the binary numbers.



```
\begin{tikzpicture}[scale=.55]
  \GraphInit[vstyle=Art]
  \SetVertexNoLabel
  \tikzset{VertexStyle/.append style = {minimum size = 5pt}}
  \def\L{2.35}
  \def\cx{{0,0.5,1}}\def\cy{{0,0.866,0}}
  \foreach \a in {0,1,2}{\foreach \b in {0,1,2}{%
  \foreach \c in {0,1,2}{\foreach \d in {0,1,2}{%
    \pgfmathsetmacro\x{\cx[\a]*\L^3+\cx[\b]*\L^2+\cx[\c]*\L+\cx[\d]}
    \pgfmathsetmacro\y{\cy[\a]*\L^3+\cy[\b]*\L^2+\cy[\c]*\L+\cy[\d]}
    \Vertex[x=\x,y=\y]{\a\b\c\d}}}}}
  \foreach \i/\j in {0/1,0/2,1/2}{%
    \foreach \a in {0,1,2}{\foreach \b in {0,1,2}{\foreach \c in {0,1,2}{%
      \Edge[local,color=gray,lw=.6pt](\a\b\c\i)(\a\b\c\j)}}}
    \foreach \a in {0,1,2}{\foreach \b in {0,1,2}{%
      \Edge[local,color=blue,lw=.8pt](\a\b\i\j)(\a\b\j\i)}}
    \foreach \a in {0,1,2}{%
      \Edge[local,color=green!60!black,lw=1pt](\a\i\j\j)(\a\j\i\i)}
    \Edge[local,color=orange,lw=1.2pt](\i\j\j\j)(\j\i\i\i)}
  \tkzHighlightPath[color=red,lw=2.5pt,style={opacity=.6}]%
    {0000,0002,0020,0022,0200,0202,0220,0222,
     2000,2002,2020,2022,2200,2202,2220,2222}
\end{tikzpicture}
```

4.3.9 The Hoffman--Singleton graph

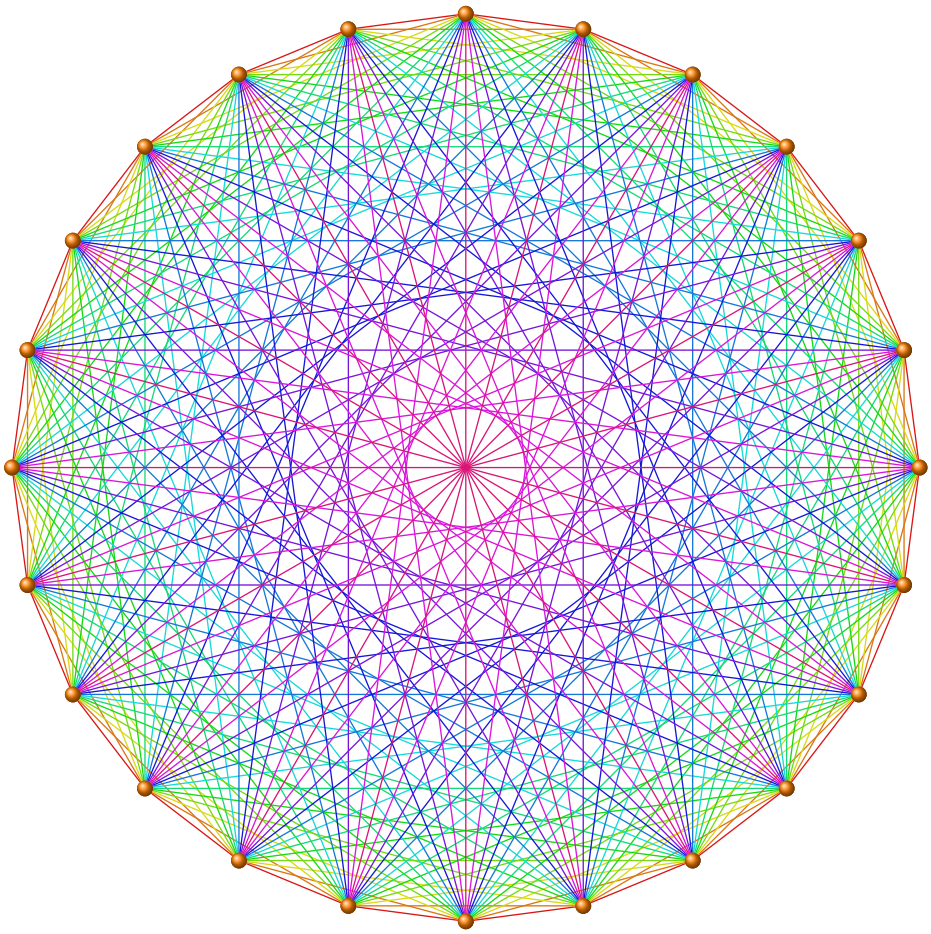
50 vertices, 175 edges, 7-regular, girth 5 : a Moore graph. Robertson's construction : five pentagons P_h, five pentagrams Q_i, and P_{hj} is adjacent to Q_{ik} when $k = hi + j \pmod 5$. The color of an edge gives h.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \SetVertexNoLabel
  \tikzset{VertexStyle/.append style = {minimum size = 6pt}}
  \def\R{6}\def\r{.9}
  \foreach \h in {0,...,4}{\foreach \j in {0,...,4}{%
    \pgfmathsetmacro\x{\R*cos(72*\h+90)+\r*cos(72*\j+90)}
    \pgfmathsetmacro\y{\R*sin(72*\h+90)+\r*sin(72*\j+90)}
    \Vertex[x=\x,y=\y]{P\h\j}
    \pgfmathsetmacro\x{(\R-2.2)*cos(72*\h+126)+\r*cos(72*\j+90)}
    \pgfmathsetmacro\y{(\R-2.2)*sin(72*\h+126)+\r*sin(72*\j+90)}
    \Vertex[x=\x,y=\y]{Q\h\j}}}
  \foreach \h/\c in {0/red,1/orange,2/green!60!black,3/blue,4/violet}{%
    \foreach \i in {0,...,4}{\foreach \j in {0,...,4}{%
      \pgfmathtruncatemacro\k{mod(\h*\i+\j,5)}
      \Edge[local,color=\c!70,lw=.4pt](P\h\j)(Q\i\k)}}}
  \foreach \h in {0,...,4}{%
    \Edges[local,lw=1.2pt](P\h0,P\h1,P\h2,P\h3,P\h4,P\h0)
    \Edges[local,lw=1.2pt](Q\h0,Q\h2,Q\h4,Q\h1,Q\h3,Q\h0)}
  \foreach \h in {0,...,4}{\foreach \j in {0,...,4}{%
    \Vertex[Node]{P\h\j}\Vertex[Node]{Q\h\j}}}
\end{tikzpicture}
```

4.3.10 The mystic rose

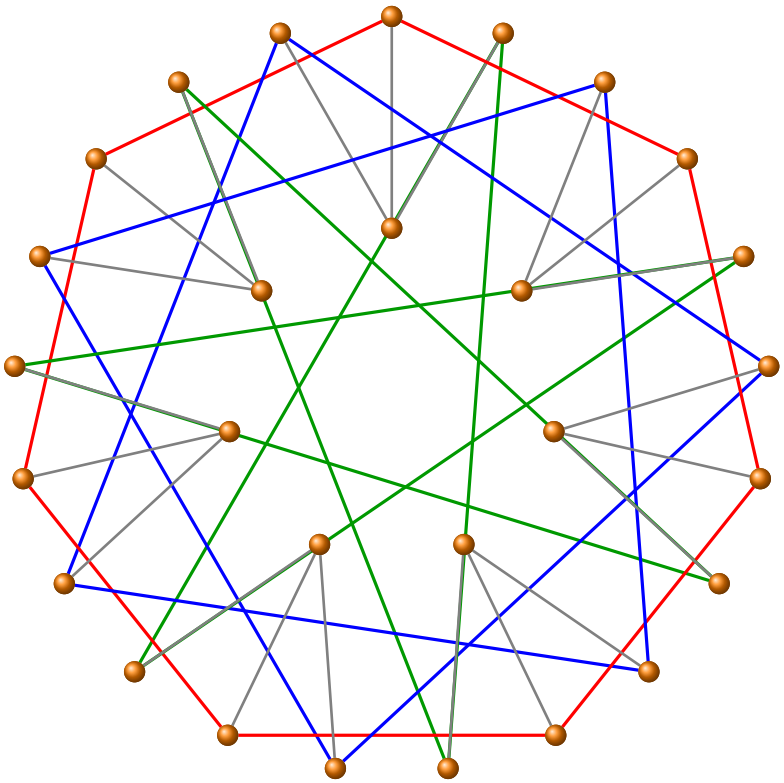
The complete graph K_{24} ; the color of an edge (hue of the model `hsb`) gives the distance between its ends along the circle.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \SetVertexNoLabel
  \tikzset{VertexStyle/.append style = {minimum size = 6pt}}
  \grEmptyCycle[RA=6]{24}
  \foreach \d in {1,...,12}{%
    \pgfmathsetmacro\hue{(\d-1)/12}
    \definecolor{h}{hsb}{\hue,0.9,0.85}
    \colorlet{c}[rgb]{h}
    \pgfmathtruncatemacro\m{\d==12 ? 11 : 23}
    \foreach \i in {0,...,\m}{%
      \pgfmathtruncatemacro\j{mod(\i+\d,24)}
      \Edge[local,color=c,lw=.5pt](a\i)(a\j)}}
  \grEmptyCycle[RA=6]{24}
\end{tikzpicture}
```

4.3.11 The Coxeter graph

28 vertices, 42 edges, cubic, girth 7, not Hamiltonian. Three 7-cycles a (step 1), b (step 2) and c (step 3) on the same circle, and d_i joined to a_i, b_i and c_i.



```
\begin{tikzpicture}
  \GraphInit[vstyle=Art]
  \SetVertexNoLabel
  \tikzset{VertexStyle/.append style = {minimum size = 8pt}}
  \foreach \i in {0,...,6}{%
    \pgfmathsetmacro\t{90+360*\i/7}
    \Vertex[a=\t,d=5]{a\i}
    \Vertex[a=\t+360/21,d=5]{b\i}
    \Vertex[a=\t-360/21,d=5]{c\i}
    \Vertex[a=\t,d=2.2]{d\i}}
  \foreach \i in {0,...,6}{%
    \pgfmathtruncatemacro\p{mod(\i+1,7)}
    \pgfmathtruncatemacro\q{mod(\i+2,7)}
    \pgfmathtruncatemacro\r{mod(\i+3,7)}
    \Edge[local,color=red,lw=1.2pt](a\i)(a\p)
    \Edge[local,color=blue,lw=1.2pt](b\i)(b\q)
    \Edge[local,color=green!60!black,lw=1.2pt](c\i)(c\r)
    \foreach \v in {a,b,c}{\Edge[local,color=gray,lw=1pt](d\i)(\v\i)}}
  \foreach \i in {0,...,6}{\foreach \v in {a,b,c,d}{\Vertex[Node]{\v\i}}}
\end{tikzpicture}
```

5 Macros and Styles

5.1 How to change the background color and text color

You can use the following macro :

`\tkzSetUpColors[⟨local options⟩]`

Options	default	definition
background	white	couleur du fond
text	black	couleur du texte

5.2 Modification of labels \AssignVertexLabel

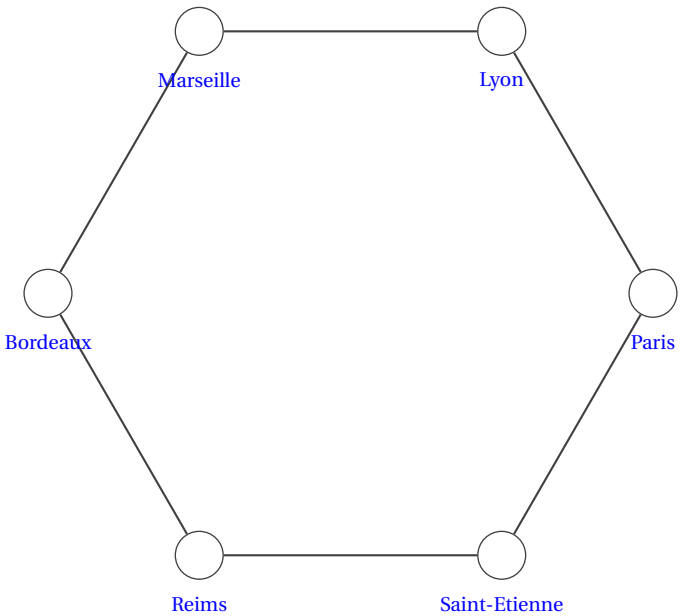
`\AssignVertexLabel[⟨local options⟩]{⟨prefix⟩}{⟨List of names⟩}`

Arguments	example
prefix	\AssignVertexLabel{a}{Alter}
List of names	\AssignVertexLabel{a}{Paris,Lyon}

Options	default	definition
size	\normalsize	taille de la fonte
color	black	couleur du texte
Math	false	math mode

5.2.1 AssignStyle and \AssignVertexLabel

First step : We create an empty graph without labels.
Second step : We place labels with the macro `\AssignVertexLabel`



```
\begin{tikzpicture}
  \SetVertexNoLabel
  \grCycle{6}
  \tikzset{AssignStyle/.append style = {below=12pt}}
  \AssignVertexLabel[color = blue,%
                     size  = \footnotesize]{a}{%
   Paris,Lyon,Marseille,Bordeaux,Reims,Saint-Etienne}
\end{tikzpicture}
```

Index

\AssignVertexLabel{a}{Alter}, 69
\AssignVertexLabel{a}{Paris,Lyon}, 69
\AssignVertexLabel, 6, 69
\AssignVertexLabel: arguments
    List of names, 69
    prefix, 69
\AssignVertexLabel: options
    Math, 69
    color, 69
    size, 69

\EdgeDoubleMod, 6, 30
\EdgeFromOneToAll, 6, 23
\EdgeFromOneToComp, 6, 26
\EdgeFromOneToSel, 6, 25
\EdgeFromOneToSeq, 6, 24
\EdgeIdentity*, 6, 21
\EdgeIdentity, 6, 20
\EdgeInGraphFromOneToComp, 6, 12
\EdgeInGraphFromOneToComp: arguments
    order, 12
\EdgeInGraphFromOneToComp: options
    Math, 12
    RA, 12
    prefix, 12
\EdgeInGraphLoop*, 14
\EdgeInGraphLoop, 6, 13
\EdgeInGraphMod*, 6, 17
\EdgeInGraphMod, 6, 16
\EdgeInGraphModLoop, 6, 18
\EdgeInGraphSeq, 6, 15
\EdgeMod*, 6, 28
\EdgeMod, 6, 27

\grAndrasfai, 6
\grBalaban, 6
\grChvatal, 6
\grCirculant, 6, 35, 61
\grCocktailParty, 6
\grComplete, 6, 33
\grCompleteBipartite, 6, 45
\grCompleteBipartite: options
    Math , 45
    RA , 45
    RB , 45
    RS , 45
    form , 45
    prefix , 45
    prefixx, 45
\grCrown, 6
\grCubicalGraph, 6
\grCycle, 6, 32
\grDesargues, 6
\grDodecahedral, 6
\grDoyle, 6
\grEmptyCycle, 6, 7

\grEmptyCycle: arguments
order, 7
\grEmptyCycle: options
Math, 7
RA, 7
prefix, 7
\grEmptyGrid, 6
\grEmptyLadder, 6
\grEmptyPath, 6, 9
\grEmptyPath: arguments
order, 9
\grEmptyPath: options
Math, 9
RA, 9
RS, 9
prefix, 9
\grEmptyStar, 6
\grFolkman, 6
\grFoster, 6
\grFranklin, 6
\grGeneralizedPetersen, 6
\grGrotzsch, 6, 59
\grHeawood, 6
\grHypercube, 6, 53, 62
\grIcosahedral, 6
\grKonisberg, 6
\grLadder, 6, 42
\grLadder: options
Math , 42
RA , 42
RS , 42
prefix , 42
prefixx, 42
\grLCF, 6, 50
\grLevi, 6
\grMcGee, 6
\grMobiusKantor, 6
\grMobiusLadder, 6
\grOctahedral, 6
\grPappus, 6
\grPath, 6
\grPetersen, 6
\grPrism, 6, 43
\grPrism: options
Math , 43
RA , 43
RB , 43
prefix , 43
prefixx , 43
\grRobertson, 6
\grRobertsonWegner, 6
\grSQCycle, 6, 40
\grStar, 6, 39, 40
\grTetrahedral, 6
\grTree, 6, 53
\grTriangularGrid, 6, 47
\grTriangularGrid: options
Math , 47

RA , 47
form , 47
prefix, 47
\grTutteCoxeter, 6
\grWheel, 6, 41
\grWong, 6
\grWriteExplicitLabel, 6
\grWriteExplicitLabels, 6

\normalsize, 69

\pgfmathtruncatemacro, 56

\SetVertexMath, 8
\SetVertexNoLabel, 7

\tkzGraphEuler, 60
\tkzSetUpColors, 69
\tkzSetUpColors: options
background, 69
text, 69