Graph Theory

j00 h4>< ?

In[1]:=
```
WordData["hacker"] // TableForm
```

Out[1]//TableForm=

hacker	Noun	UnskilledPerson
hacker	Noun	Coder
hacker	Noun	GolfPlayer
hacker	Noun	Terrorist

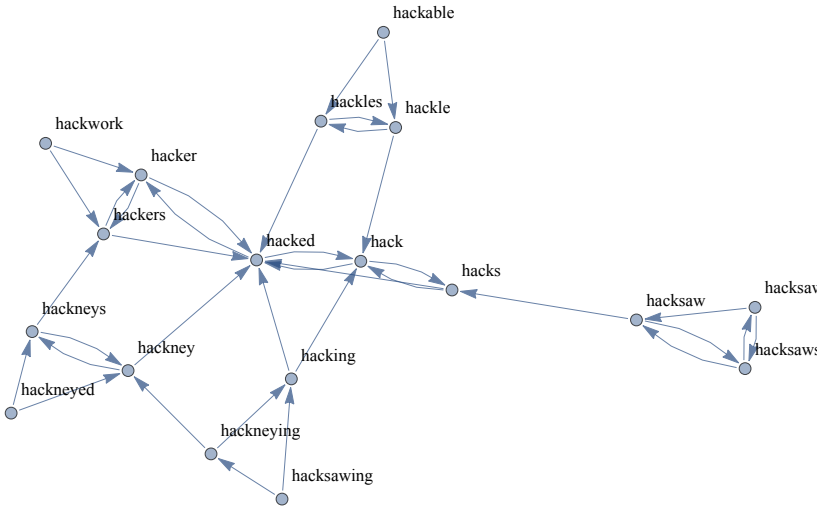
In[2]:=
```
WordData["hacker", "Definitions"] // TableForm
```

Out[2]//TableForm=

```
{hacker, Noun, UnskilledPerson} → one who works hard at boring tasks
{hacker, Noun, Coder} → a programmer for whom computing is its own reward; may enjoy
{hacker, Noun, GolfPlayer} → someone who plays golf poorly
{hacker, Noun, Terrorist} → a programmer who breaks into computer systems in order to
```

In[3]:=
```
words = DictionaryLookup["hack*"];
Flatten[Map[(Thread[# ð DeleteCases[Nearest[words, #, 3], #]]) &, words]];
Graph[%, VertexLabels → "Name", ImagePadding → 20, ImageSize → 450]
```

Out[5]=



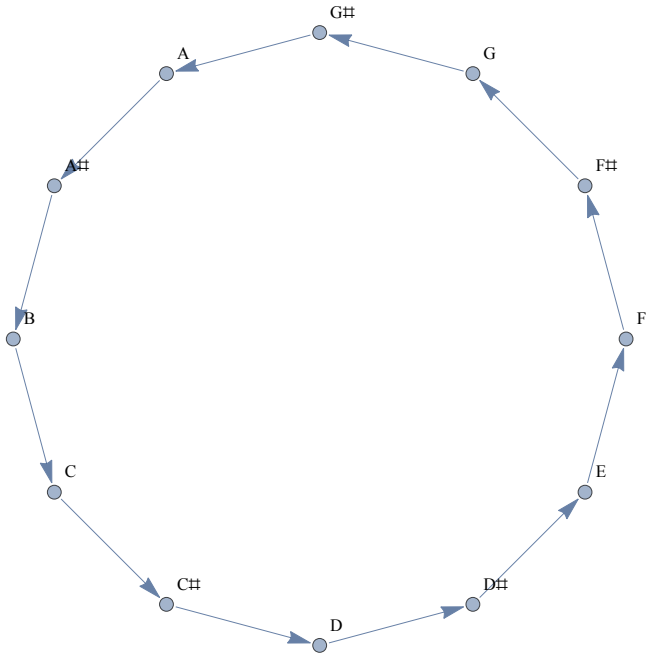
The Circle of Fifths

In[6]:=
```
fullOctave = {"A" → "A#", "A#" → "B", "B" → "C", "C" → "C#", "C#" → "D", "D" → "D#",
    "D#" → "E", "E" → "F", "F" → "F#", "F#" → "G", "G" → "G#", "G#" → "A"};
```

Printed by Wolfram Mathematica Student Edition

In[7]:=
```
Graph[fullOctave, VertexLabels → "Name",
 ImagePadding → 10, VertexShapeFunction → "Circle",
 GraphLayout → "CircularEmbedding", DirectedEdges → True]
```

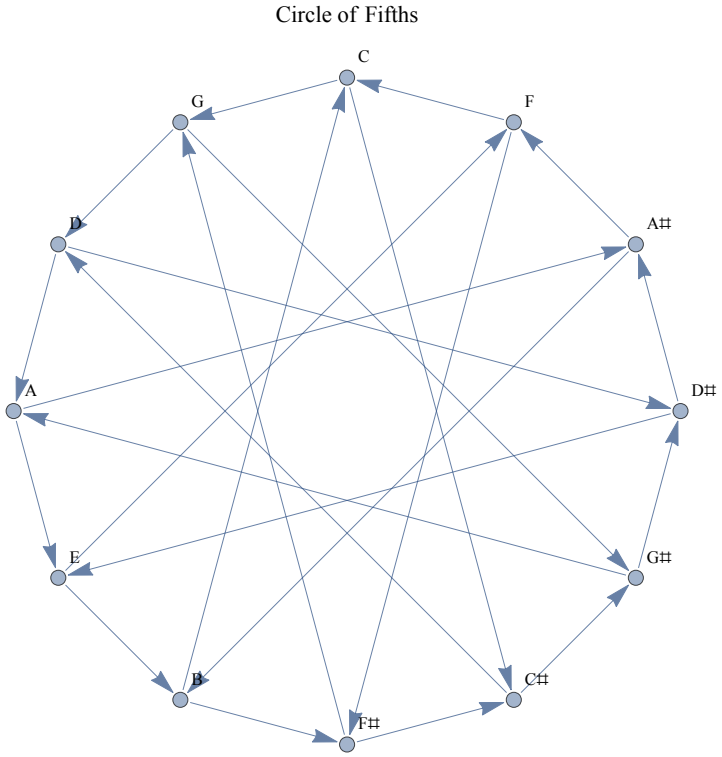
Out[7]=



In[8]:=
```
circleOfFifths = {"G" → "D", "D" → "A", "A" → "E", "E" → "B", "B" → "F♯", "F♯" → "C♯",
   "C♯" → "G♯", "G♯" → "D♯", "D♯" → "A♯", "A♯" → "F", "F" → "C", "C" → "G"};
```

In[9]:=
```
Graph[Flatten[{circleOfFifths, fullOctave}],
 VertexLabels → "Name", ImagePadding → 15, ImageSize → 400,
 VertexShapeFunction → "Circle", GraphLayout → "CircularEmbedding",
 DirectedEdges → True, PlotLabel → "Circle of Fifths"]
```

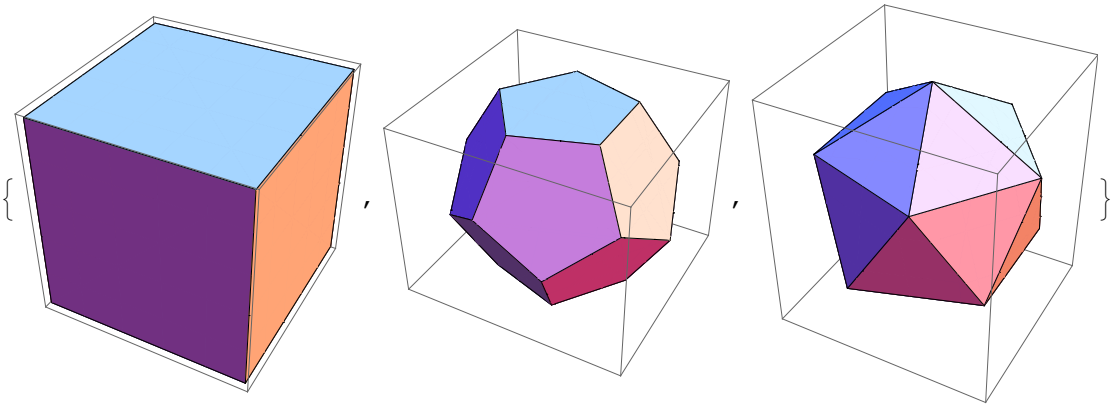
Out[9]=



Platonic Solids

In[10]:=
```
{PolyhedronData["Cube"],
 PolyhedronData["Dodecahedron"],
 PolyhedronData["Icosahedron"]}
```

Out[10]=



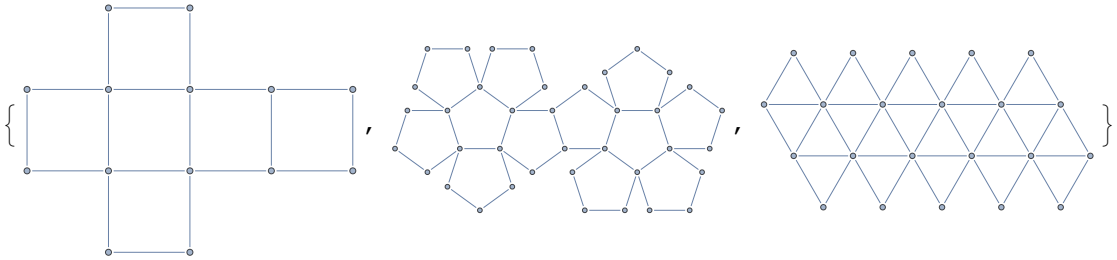
In[11]:=
```
Cube[type_] := PolyhedronData["Cube", type]
Dodeca[type_] := PolyhedronData["Dodecahedron", type]
Icosa[type_] := PolyhedronData["Icosahedron", type]

CubeNet := Cube["NetGraph"]
DodecaNet := Dodeca["NetGraph"]
IcosaNet := Icosa["NetGraph"]

CubeSkel := Cube["SkeletonGraph"]
DodecaSkel := Dodeca["SkeletonGraph"]
IcosaSkel := Icosa["SkeletonGraph"]
```

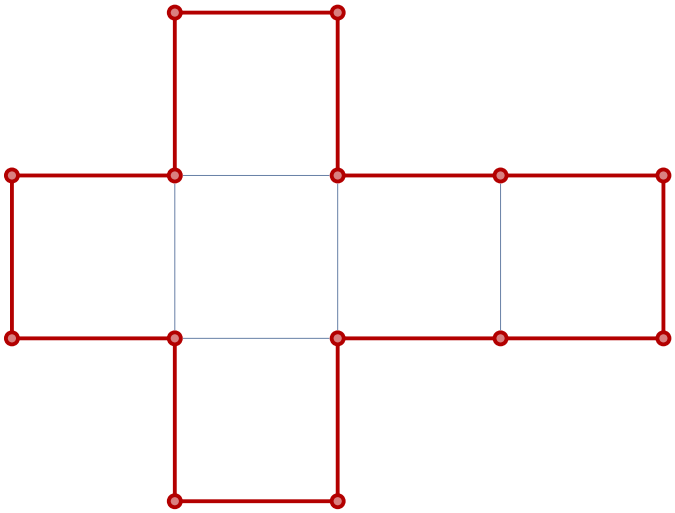
In[20]:=
```
{CubeNet, DodecaNet, IcosaNet}
```

Out[20]=

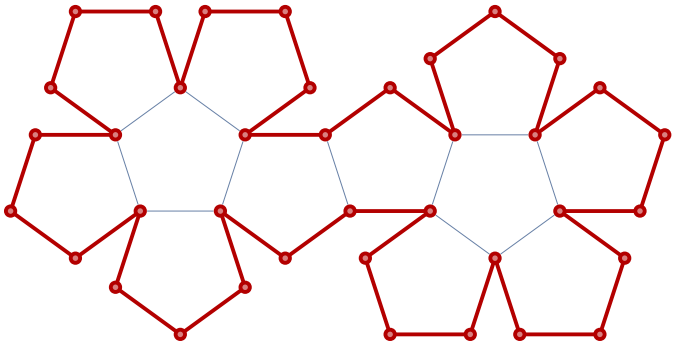


In[21]:=
```
HighlightGraph[CubeNet,
 PathGraph@First[FindHamiltonianCycle[CubeNet]], GraphHighlightStyle → "Thick"]
HighlightGraph[DodecaNet, PathGraph@First[FindHamiltonianCycle[DodecaNet]],
 GraphHighlightStyle → "Thick"]
HighlightGraph[IcosaNet, PathGraph@First[FindHamiltonianCycle[IcosaNet]],
 GraphHighlightStyle → "Thick"]
```

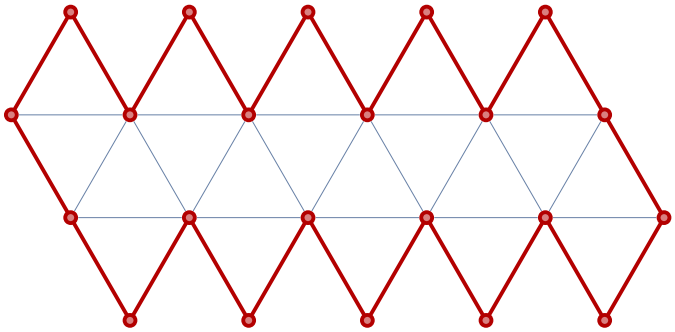
Out[21]=



Out[22]=

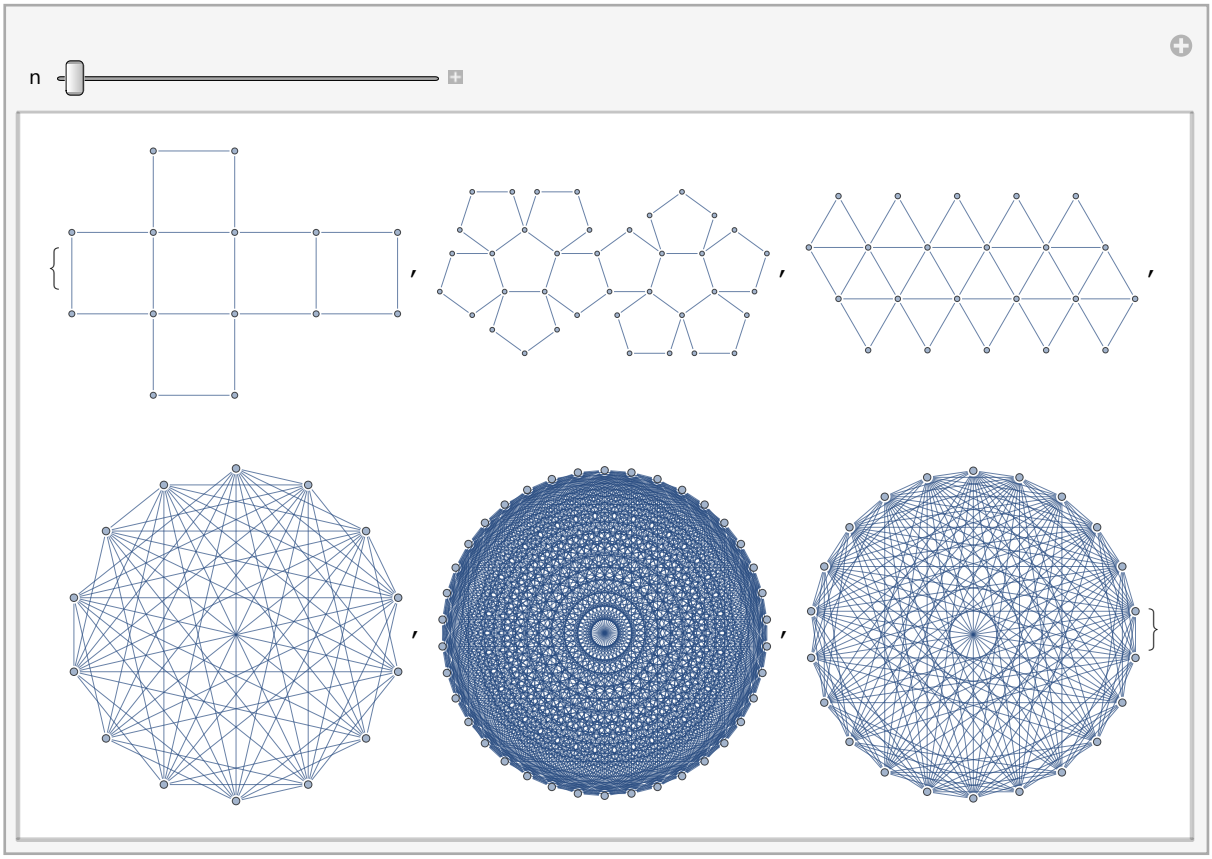


Out[23]=



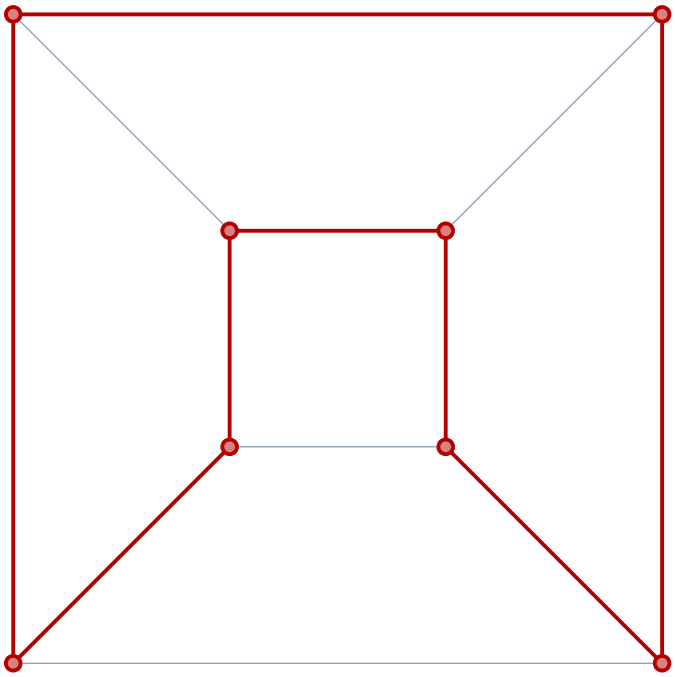
In[24]:=
```
Manipulate[{GraphPower[CubeNet, n], GraphPower[DodecaNet, n],
  GraphPower[IcosaNet, n], GraphComplement[GraphPower[CubeNet, n]],
  GraphComplement[GraphPower[DodecaNet, n]],
  GraphComplement[GraphPower[IcosaNet, n]]}, {n, 1, 8, 1}]
```

Out[24]=

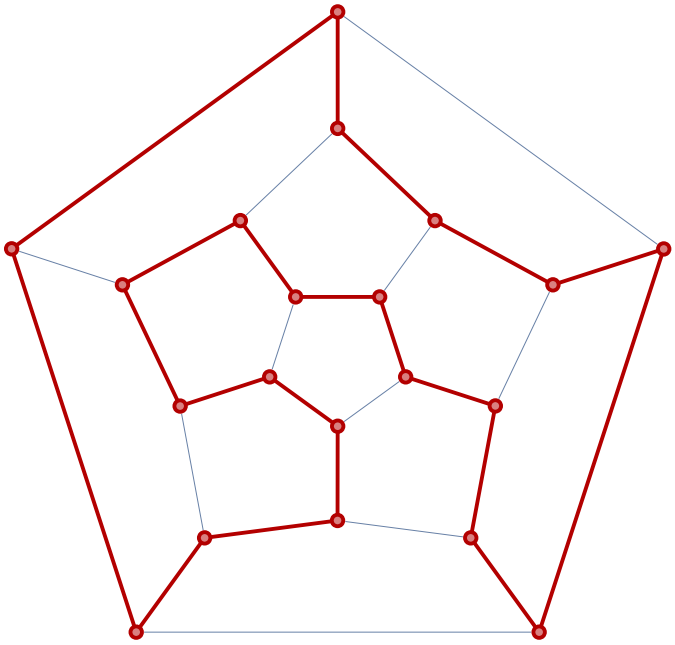


In[25]:=
```
HighlightGraph[CubeSkel,
 PathGraph@First[FindHamiltonianCycle[CubeSkel]], GraphHighlightStyle → "Thick"]
HighlightGraph[DodecaSkel, PathGraph@First[FindHamiltonianCycle[DodecaSkel]],
 GraphHighlightStyle → "Thick"]
HighlightGraph[IcosaSkel, PathGraph@First[FindHamiltonianCycle[IcosaSkel]],
 GraphHighlightStyle → "Thick"]
```

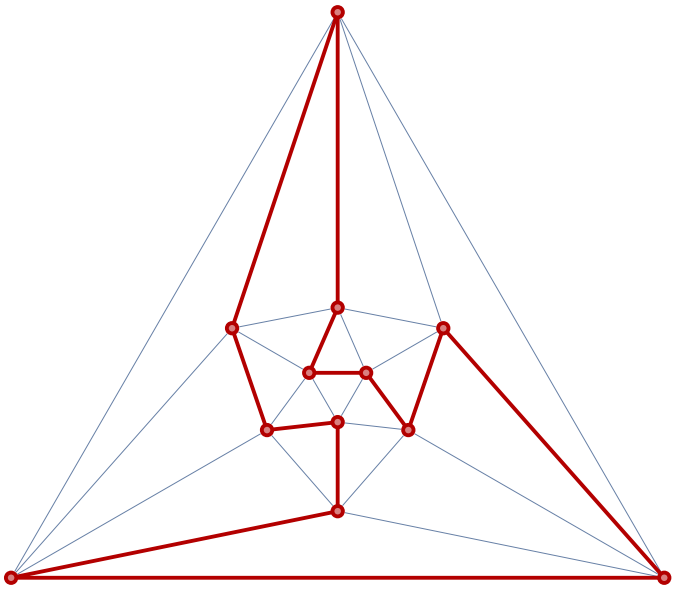
Out[25]=



Out[26]=

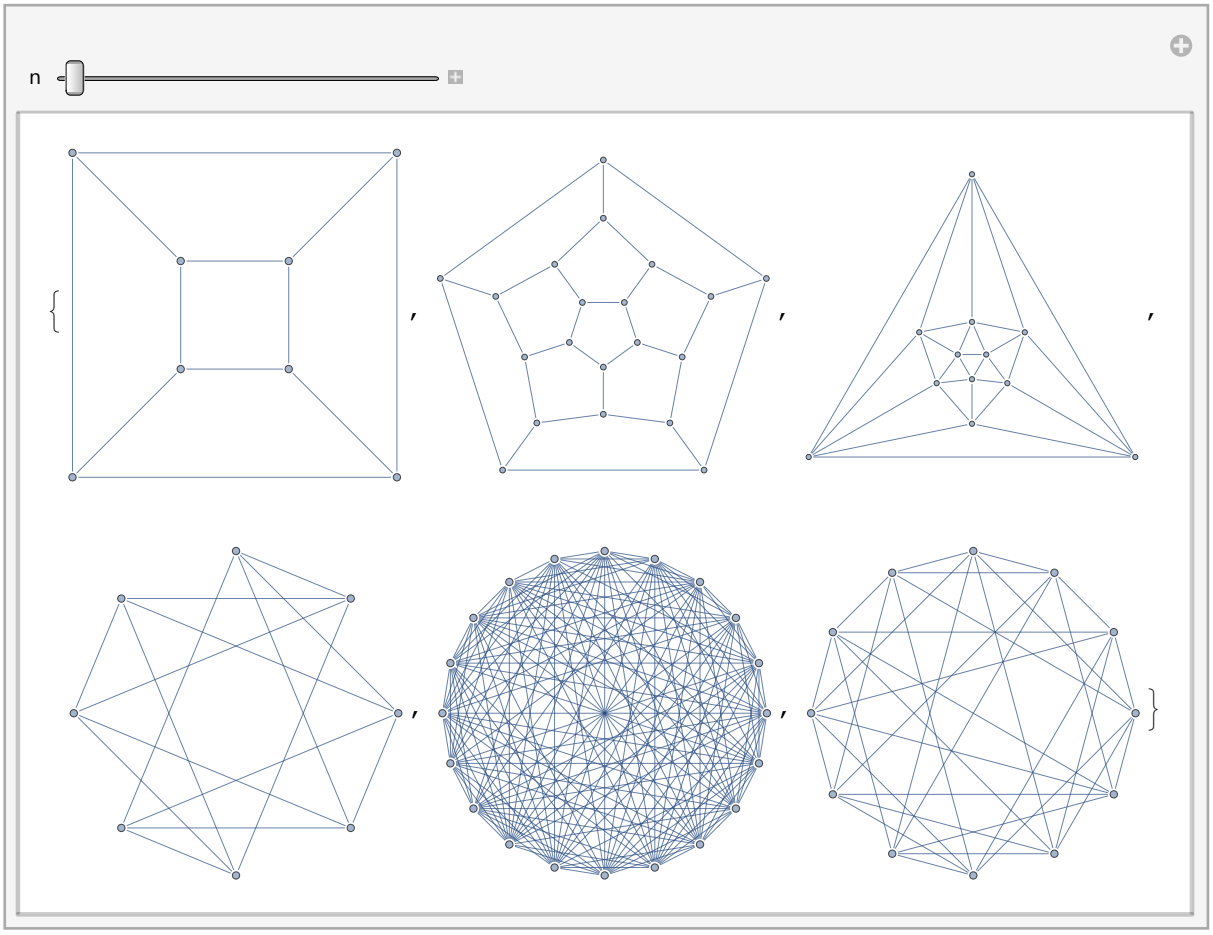


Out[27]=



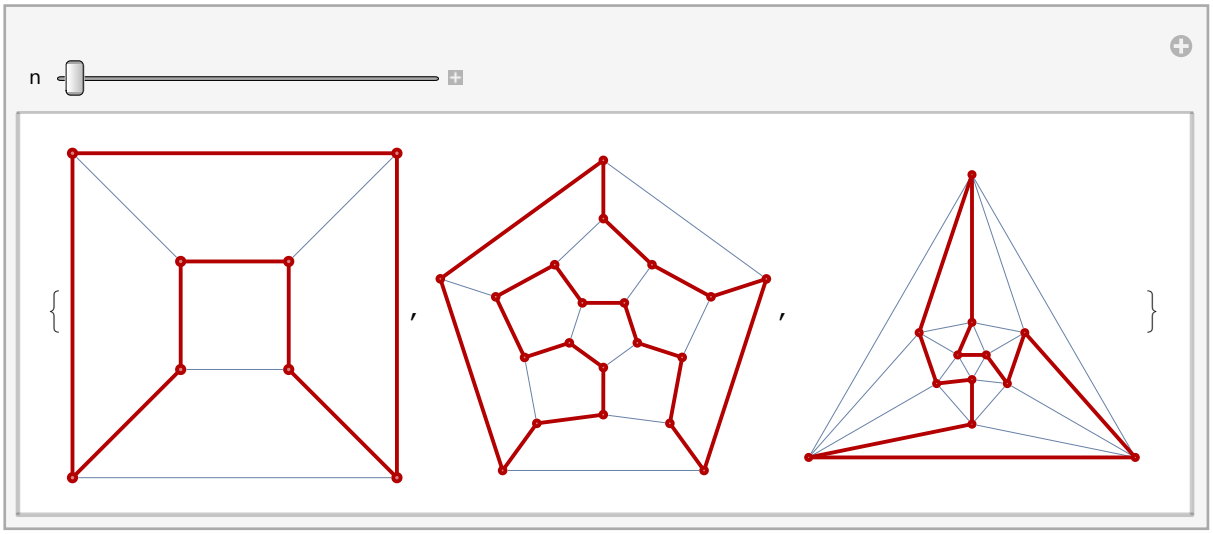
In[28]:=
```
Manipulate[{GraphPower[CubeSkel, n], GraphPower[DodecaSkel, n],
  GraphPower[IcosaSkel, n], GraphComplement[GraphPower[CubeSkel, n]],
  GraphComplement[GraphPower[DodecaSkel, n]],
  GraphComplement[GraphPower[IcosaSkel, n]]}, {n, 1, 8, 1}]
```

Out[28]=



In[29]:=
```
Manipulate[{HighlightGraph[GraphPower[CubeSkel, n],
    PathGraph@First[FindHamiltonianCycle[GraphPower[CubeSkel, n]]],
    GraphHighlightStyle → "Thick"], HighlightGraph[GraphPower[DodecaSkel, n],
    PathGraph@First[FindHamiltonianCycle[GraphPower[DodecaSkel, n]]],
    GraphHighlightStyle → "Thick"], HighlightGraph[GraphPower[IcosaSkel, n],
    PathGraph@First[FindHamiltonianCycle[GraphPower[IcosaSkel, n]]],
    GraphHighlightStyle → "Thick"]}, {n, 1, 8, 1}]
```

Out[29]=



Data Structures

Singly Linked List

In[30]:=
```
Graph[{1 → 2, 2 → 3, 3 → 4, 4 → 5, 5 → 6}, DirectedEdges → True, VertexSize → 0.2,
 EdgeStyle → Thick, VertexLabels → "Name", GraphLayout → "LinearEmbedding"]
```

Out[30]=

Doubly Linked List

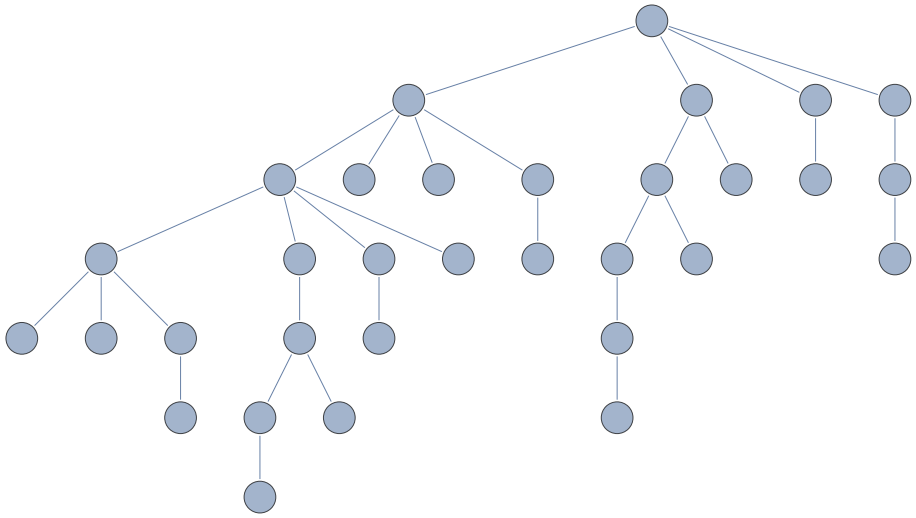
In[31]:=
```
Graph[{1 → 2, 2 → 3, 3 → 4, 4 → 5, 5 → 6, 6 → 5, 5 → 4, 4 → 3, 3 → 2, 2 → 1},
 DirectedEdges → True, VertexSize → 0.2, EdgeStyle → Thick, VertexLabels → "Name"]
```

Out[31]=

Random Tree

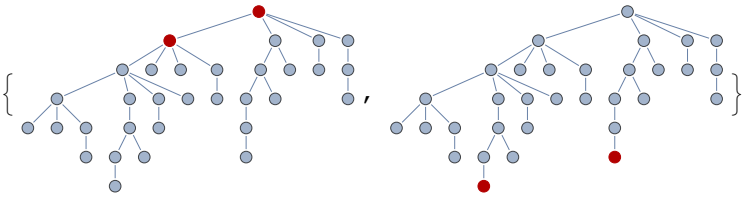
In[32]:=
```
tree = TreeGraph[RandomInteger[#] ê # + 1 & /@ Range[0, 30],
  VertexSize → 0.4, DirectedEdges → True]
```

Out[32]=



In[33]:=
```
{HighlightGraph[tree, GraphCenter[tree]],
 HighlightGraph[tree, GraphPeriphery[tree]]}
```

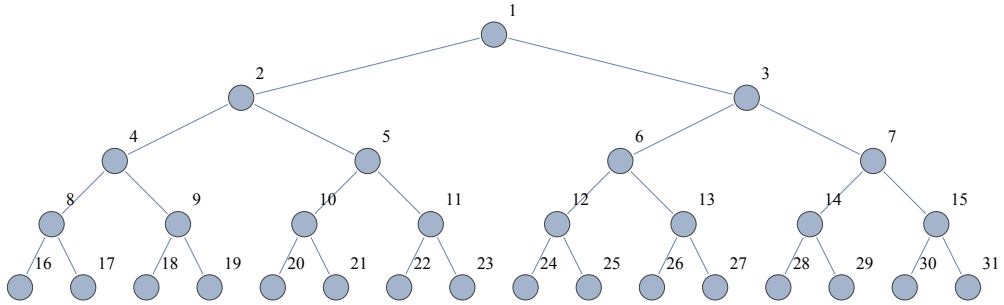
Out[33]=



Binary Tree

In[34]:=
```
b = CompleteKaryTree[5, 2, VertexLabels → "Name", VertexSize → Large]
```

Out[34]=



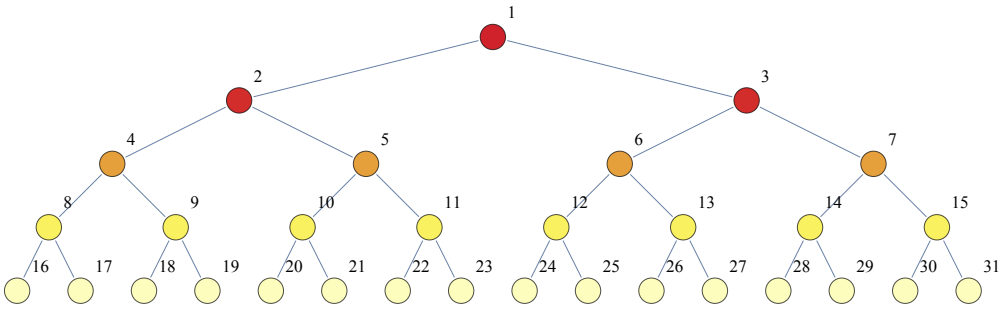
In[35]:=
```
Column[Table[BitShiftLeft[1, n], {n, 0, 6}]]
```

Out[35]=
```
1
2
4
8
16
32
64
```

In[36]:=
```
HighlightCentrality[g_, cc_] := HighlightGraph[g, Table[Style[VertexList[g][[i]],
      ColorData["TemperatureMap"][cc[[i]] / Max[cc]]], {i, VertexCount[g]}]];
```

In[37]:=
```
HighlightCentrality[b, ClosenessCentrality[b]]
```

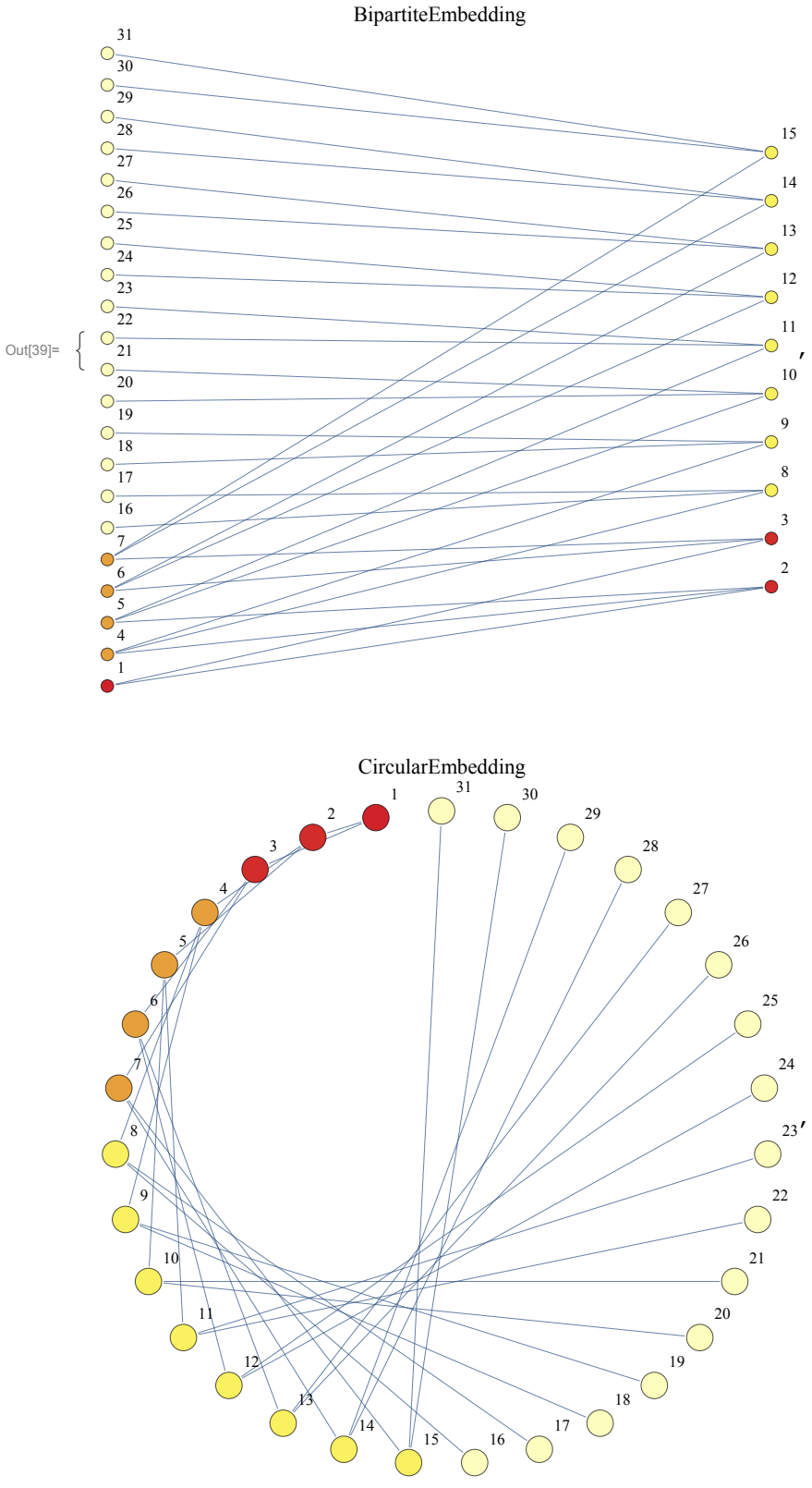
Out[37]=



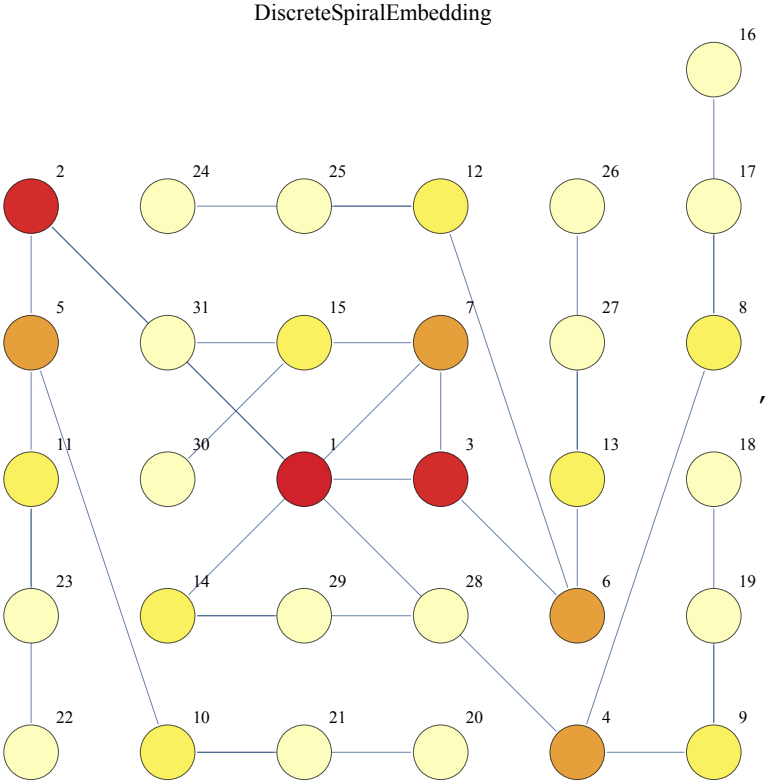
Other Types of Trees?

In[38]:=
```
bin[layout_] := CompleteKaryTree[5, 2, VertexLabels → "Name",
   VertexSize → Large, GraphLayout → layout, PlotLabel → layout, ImageSize → 400]
Table[HighlightCentrality[bin[l], ClosenessCentrality[bin[l]]],
 {l, {"BipartiteEmbedding", "CircularEmbedding", "DiscreteSpiralEmbedding",
   "GridEmbedding", "LinearEmbedding", "MultipartiteEmbedding",
   "SpiralEmbedding", "StarEmbedding", "BalloonEmbedding",
   "RadialEmbedding", "LayeredDigraphEmbedding", "LayeredEmbedding",
   "HighDimensionalEmbedding", "PlanarEmbedding", "SpectralEmbedding"}}]
```

Out[39]=


BipartiteEmbedding
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
7
6
5
4
1
15
14
13
12
11
10
9
8
3
2
CircularEmbedding
1
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2

DiscreteSpiralEmbedding

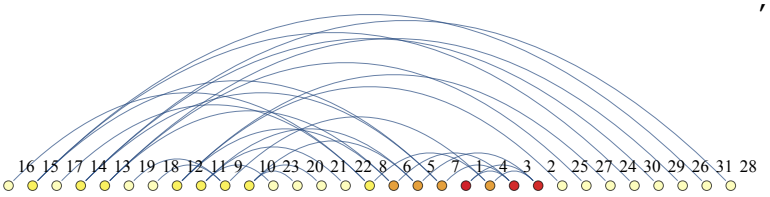


,

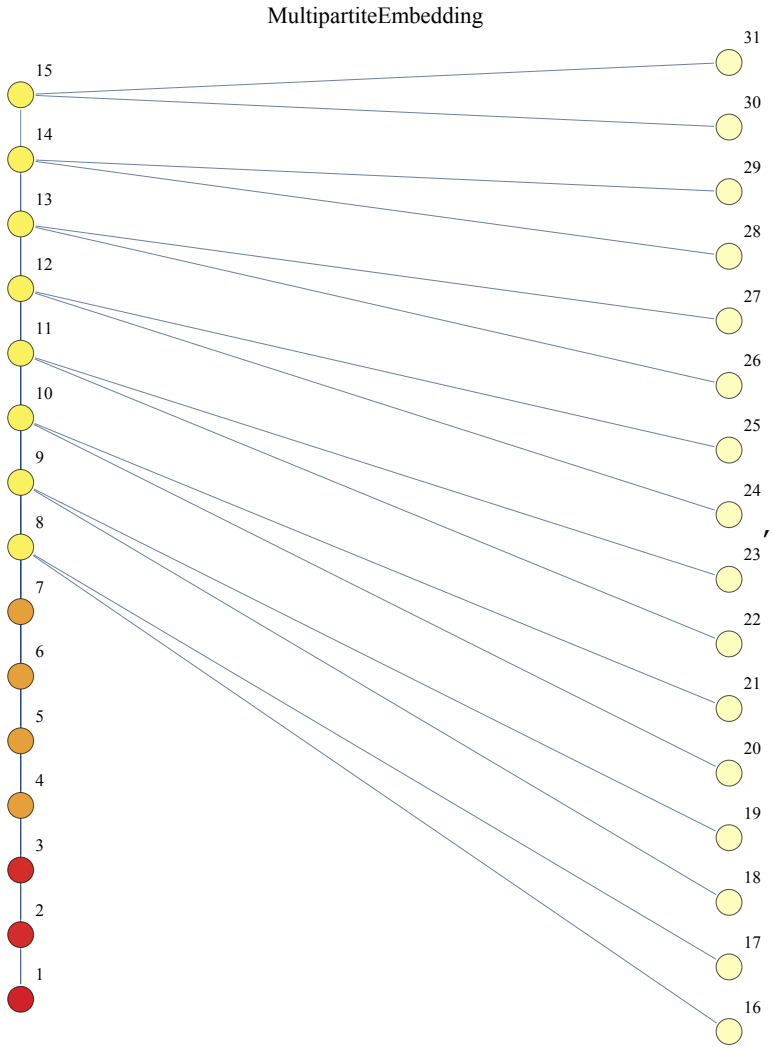
GridEmbedding

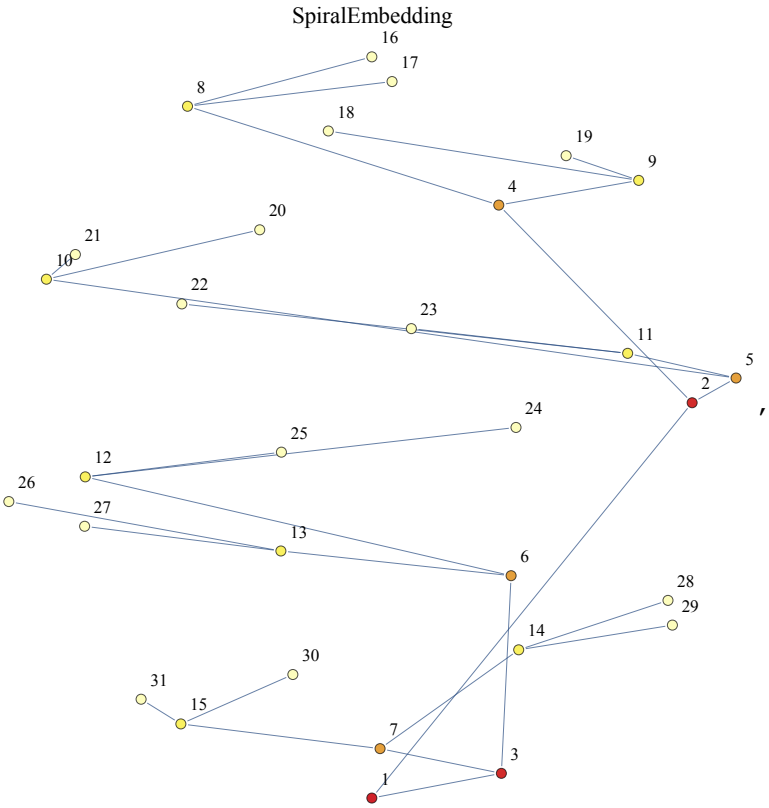
,

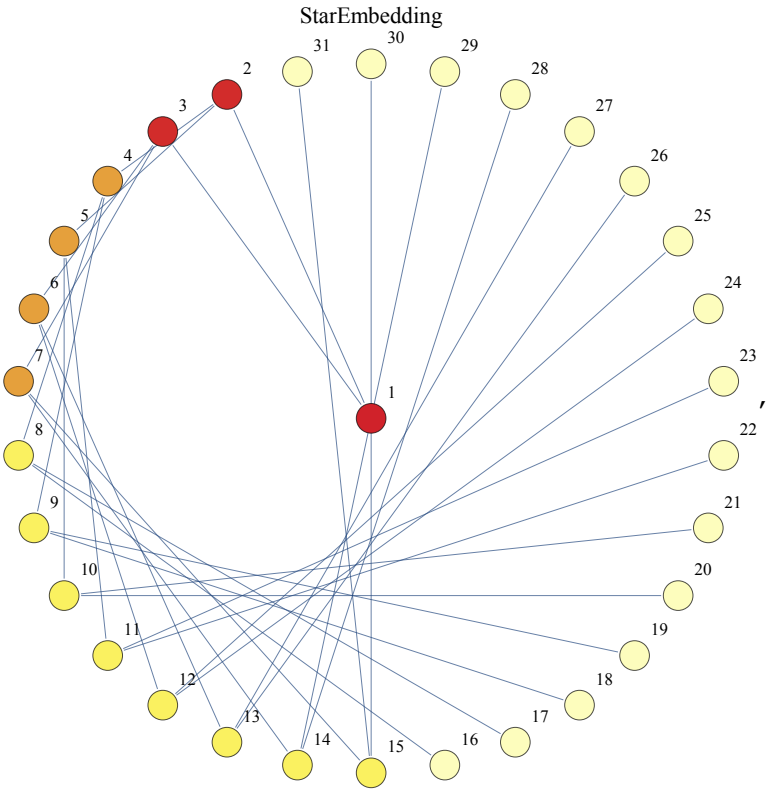
LinearEmbedding

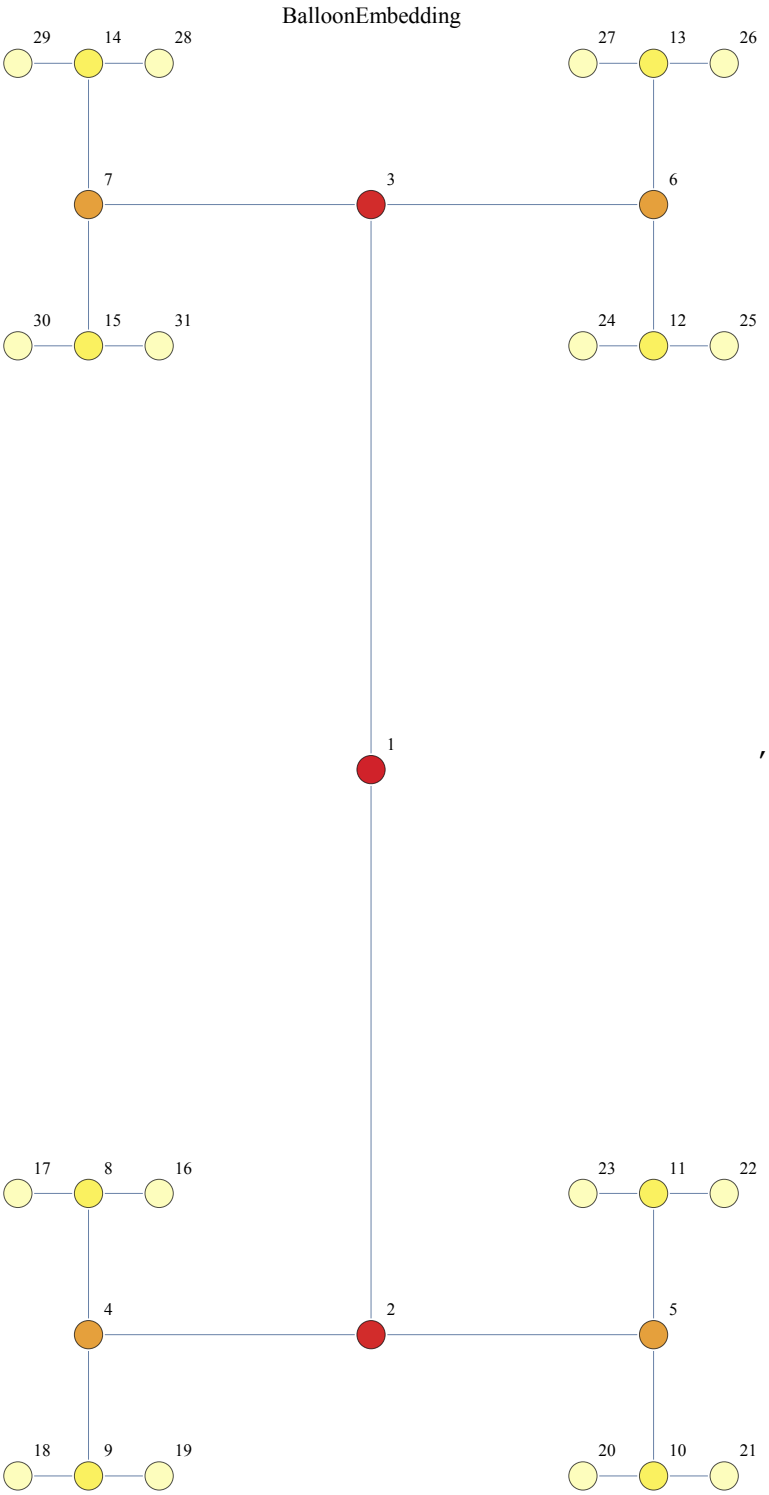


,

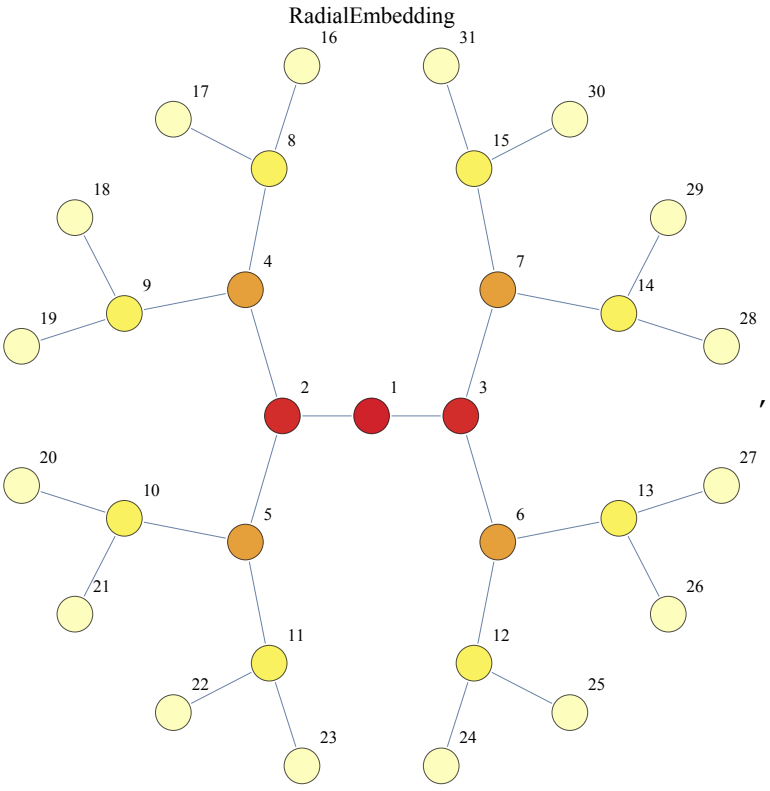

MultipartiteEmbedding
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16


SpiralEmbedding
16
17
8
18
19
9
4
20
21
10
22
23
11
5
2
24
25
12
26
27
13
6
28
29
14
30
31
15
7
3
1

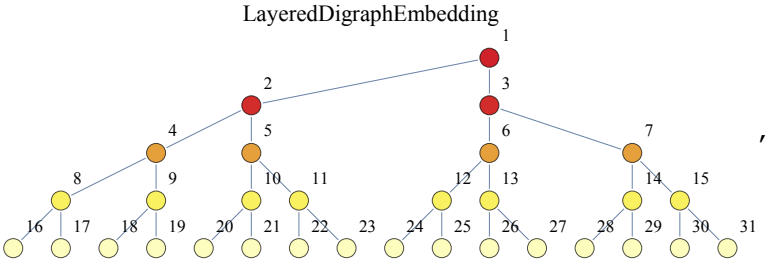

StarEmbedding
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1


BalloonEmbedding
29
14
28
27
13
26
7
3
6
30
15
31
24
12
25
1
17
8
16
23
11
22
4
2
5
18
9
19
20
10
21

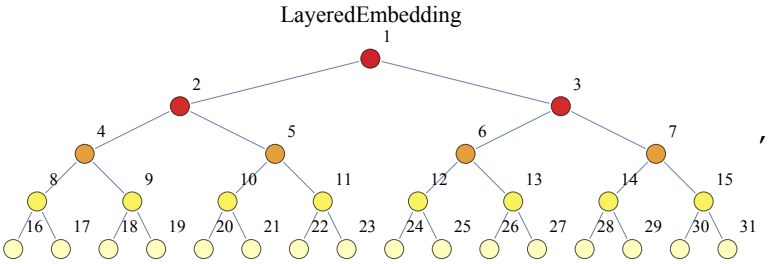
,


RadialEmbedding
16
31
17
30
8
15
18
29
4
7
9
14
19
28
2
1
3
20
27
10
13
5
6
21
26
11
12
22
25
23
24

,

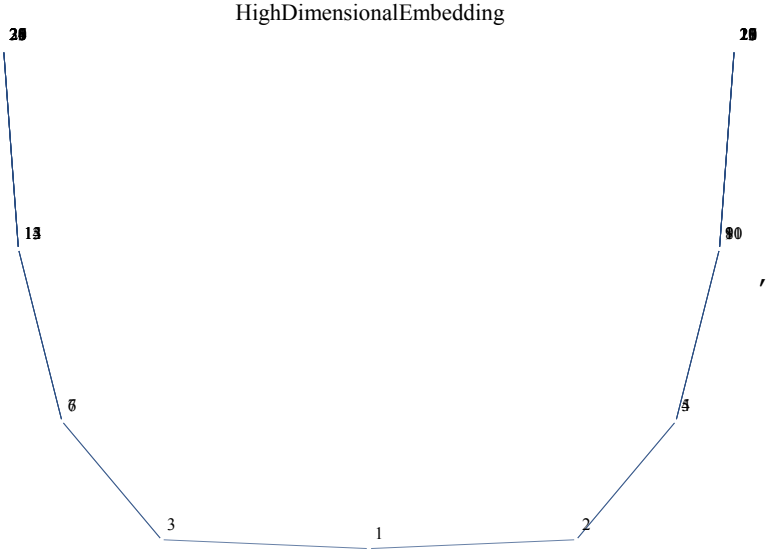

LayeredDigraphEmbedding
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

,

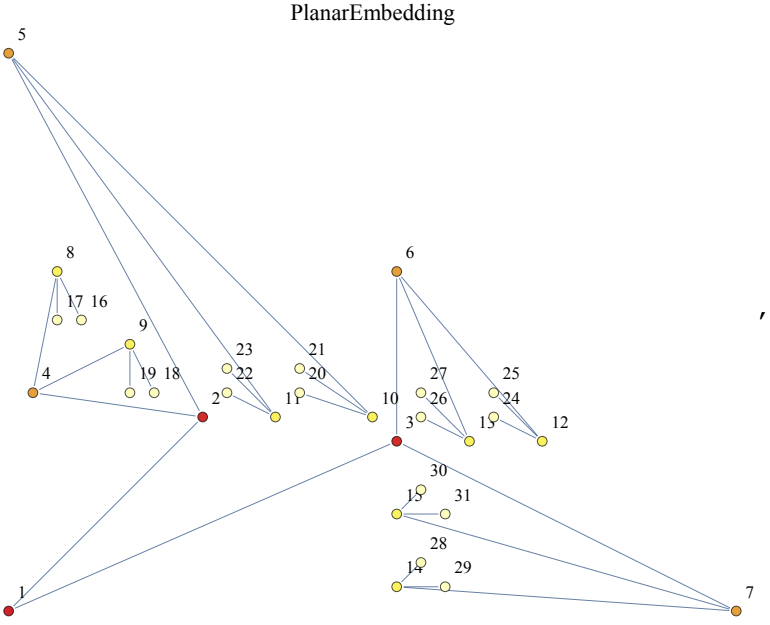

LayeredEmbedding
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

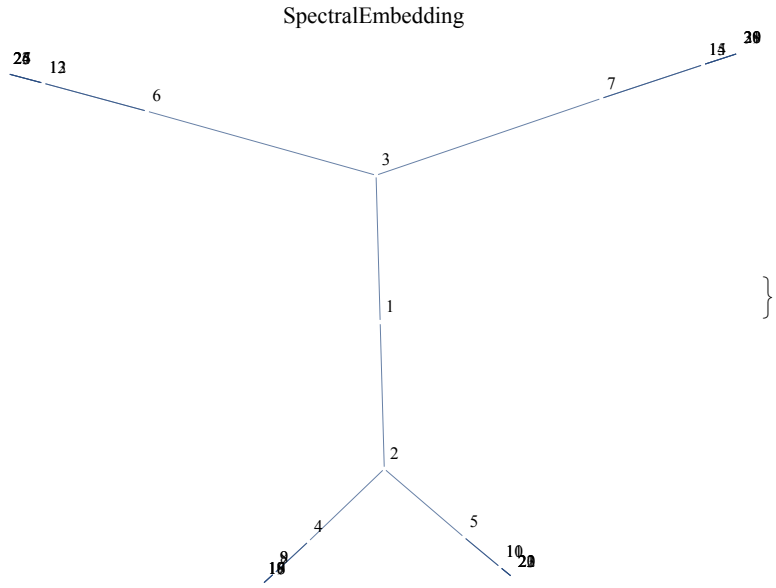
,

HighDimensionalEmbedding



PlanarEmbedding

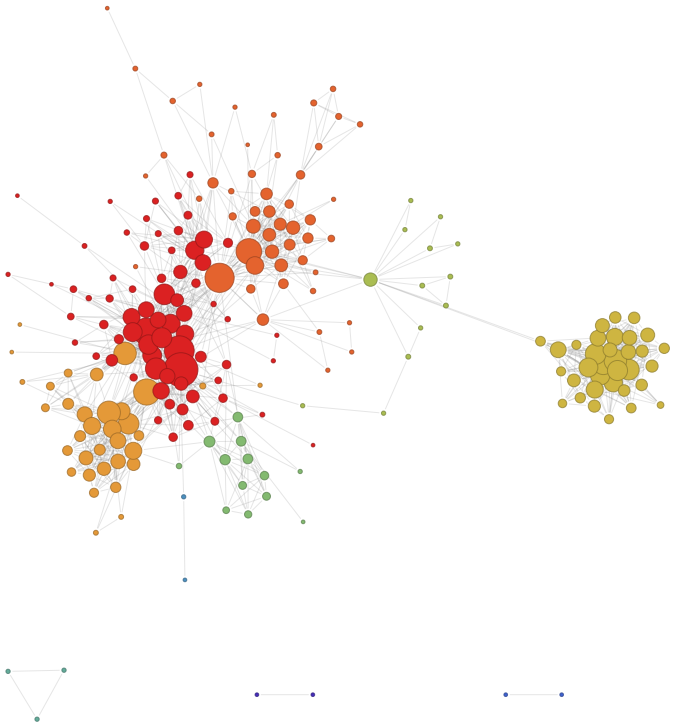




Social Media Analysis

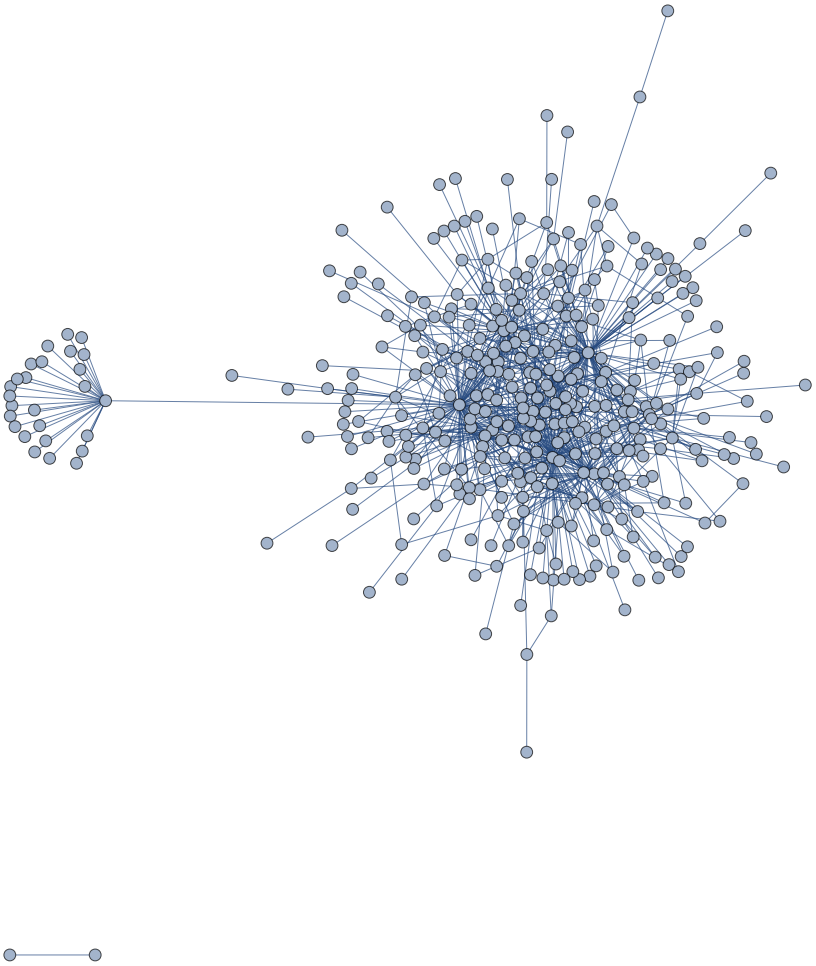
In[44]:= `SocialMediaData["Facebook", "FriendNetwork"]`

Out[44]=



In[41]:=
```
SocialMediaData["Facebook", "BimodalLikeCommentNetwork", "FormattedData"]
```

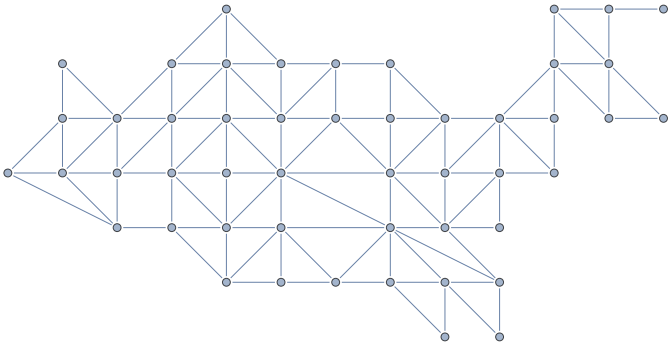
Out[41]=



Contiguous US

In[42]:=
```
GraphData["ContiguousUSAGraph"]
CountryData["UnitedStates", "Shape"]
```

Out[42]=



Out[43]=