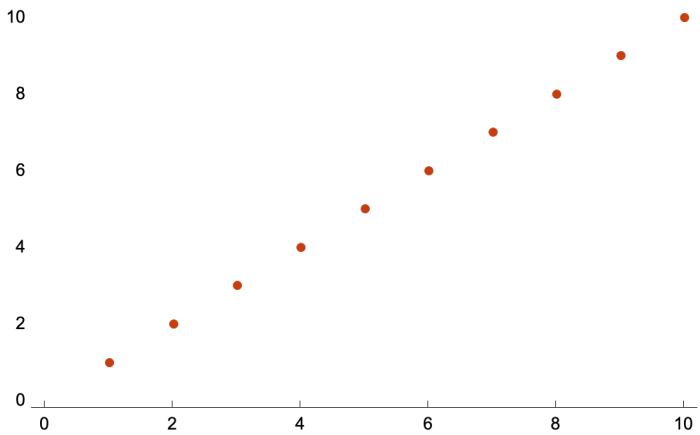


Chapter 20

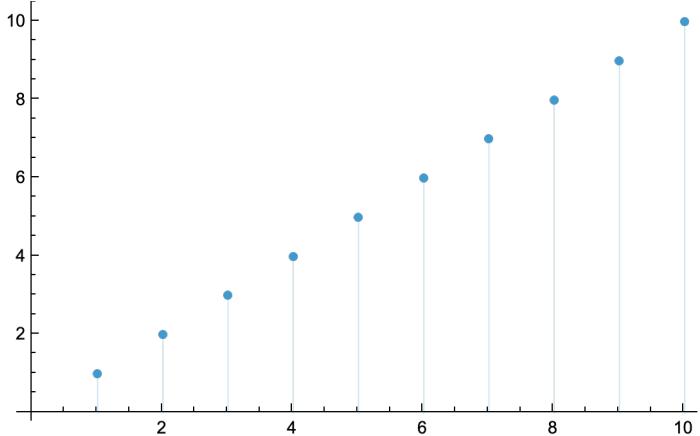
```
In[86]:= ListPlot[Range[10], PlotTheme -> "Web"]
```

```
Out[86]=
```



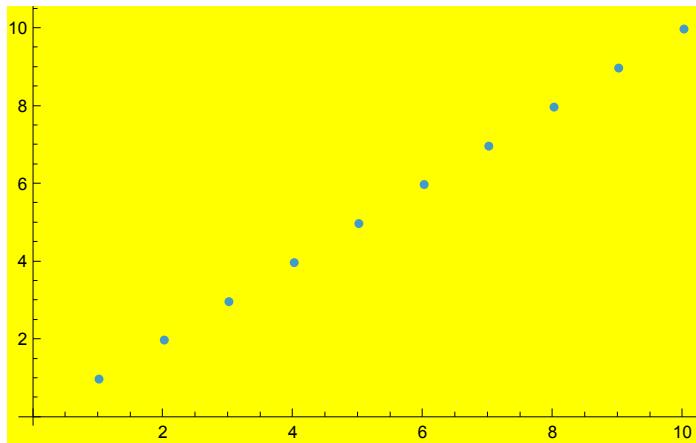
```
In[87]:= ListPlot[Range[10], Filling -> Axis]
```

```
Out[87]=
```



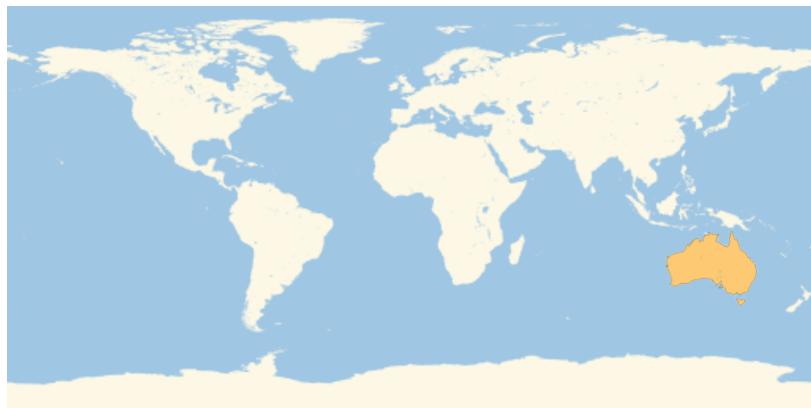
```
In[88]:= ListPlot[Range[10], Background -> Yellow]
```

```
Out[88]=
```



```
In[89]:= GeoListPlot[Entity["Country", "Australia"], GeoRange -> All]
```

```
Out[89]=
```



```
In[90]:= GeoListPlot[Entity["Country", "Madagascar"],  
GeoRange -> Entity["Ocean", "IndianOcean"]]
```

```
Out[90]=
```



```
In[91]:= GeoGraphics[EntityClass["Country", "SouthAmerica"],  
GeoBackground -> "ReliefMap"]
```

Out[91]=



```
In[92]:= GeoListPlot[{Entity["Country", "France"],  
Entity["Country", "Finland"], Entity["Country", "Greece"]},  
GeoLabels → Automatic]
```

Out[92]=



```
In[93]:= GeoListPlot[EntityClass["University", "TheIvyLeague"],  
GeoLabels → Automatic]
```

Out[93]=



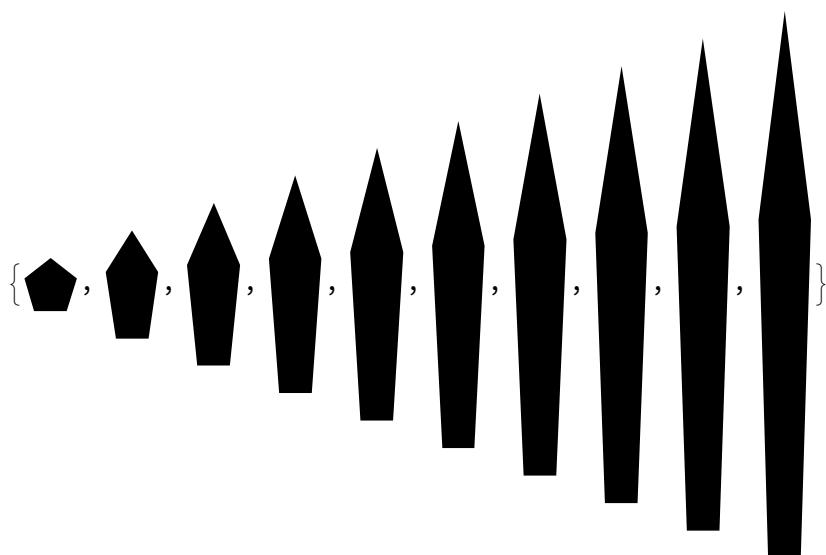
```
In[94]:= Grid[Table[Style[i j, RGBColor[1, 1, 1]], {i, 1, 12}, {j, 1, 12}],  
Background → Black]
```

Out[94]=

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

```
In[95]:= Table[Graphics[Disk[], ImageSize -> RandomInteger[40]], 100]
Out[95]= {
```

```
In[96]:= Table[Graphics[RegularPolygon[5], ImageSize -> 30,
AspectRatio -> n], {n, 1, 10}]
Out[96]= {
```

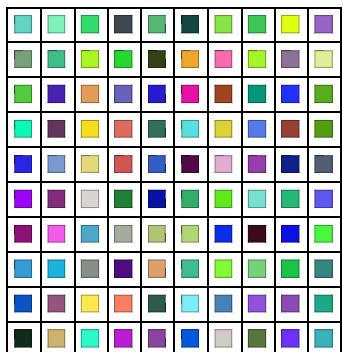


```
In[97]:= Manipulate[Graphics[Circle[], ImageSize -> n], {n, 5, 500}]
Out[97]= 
```



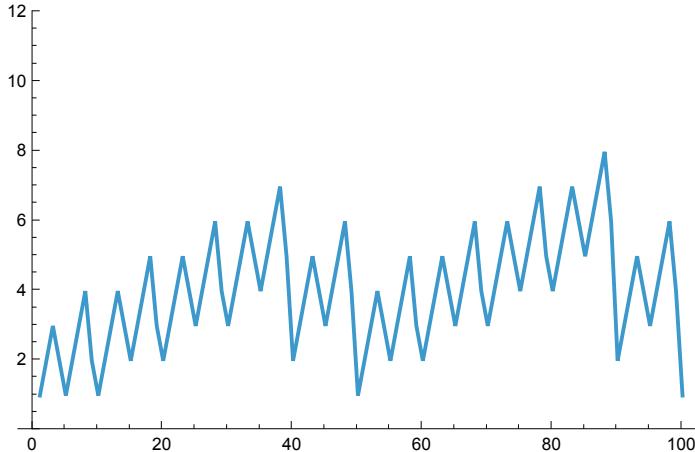
```
In[98]:= Grid[Table[RandomColor[], 10, 10], Frame -> All]
```

```
Out[98]=
```



```
In[99]:= ListLinePlot[Table[Length[Characters[RomanNumeral[n]]], {n, 1, 100}],  
PlotRange ->  
Max[Table[Length[Characters[RomanNumeral[n]]], {n, 1000}]]]
```

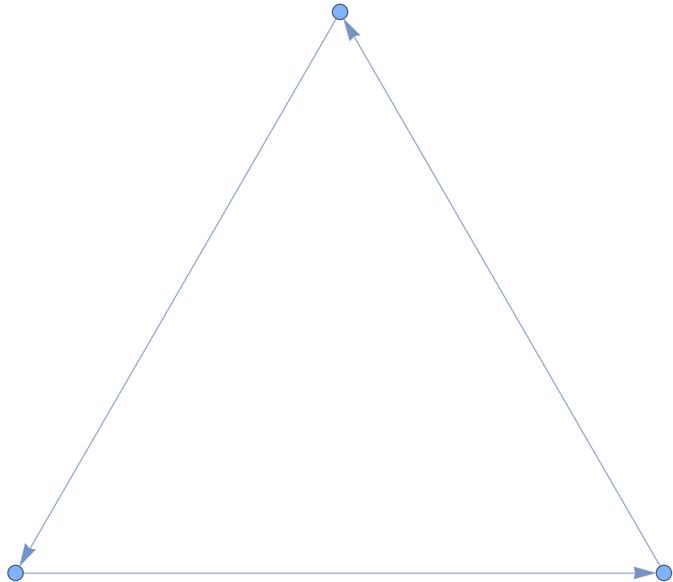
```
Out[99]=
```



Chapter 21

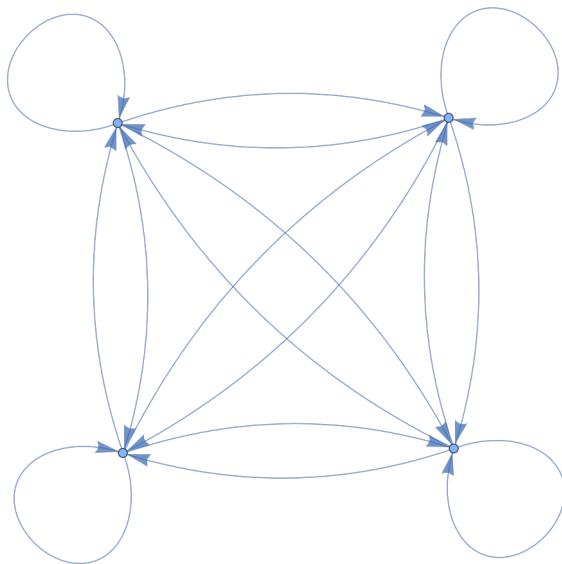
```
In[100]:= Graph[{1 → 2, 2 → 3, 3 → 1}]
```

```
Out[100]=
```



```
In[101]:= Graph[Flatten[Table[i → j, {i, 4}, {j, 4}]]]
```

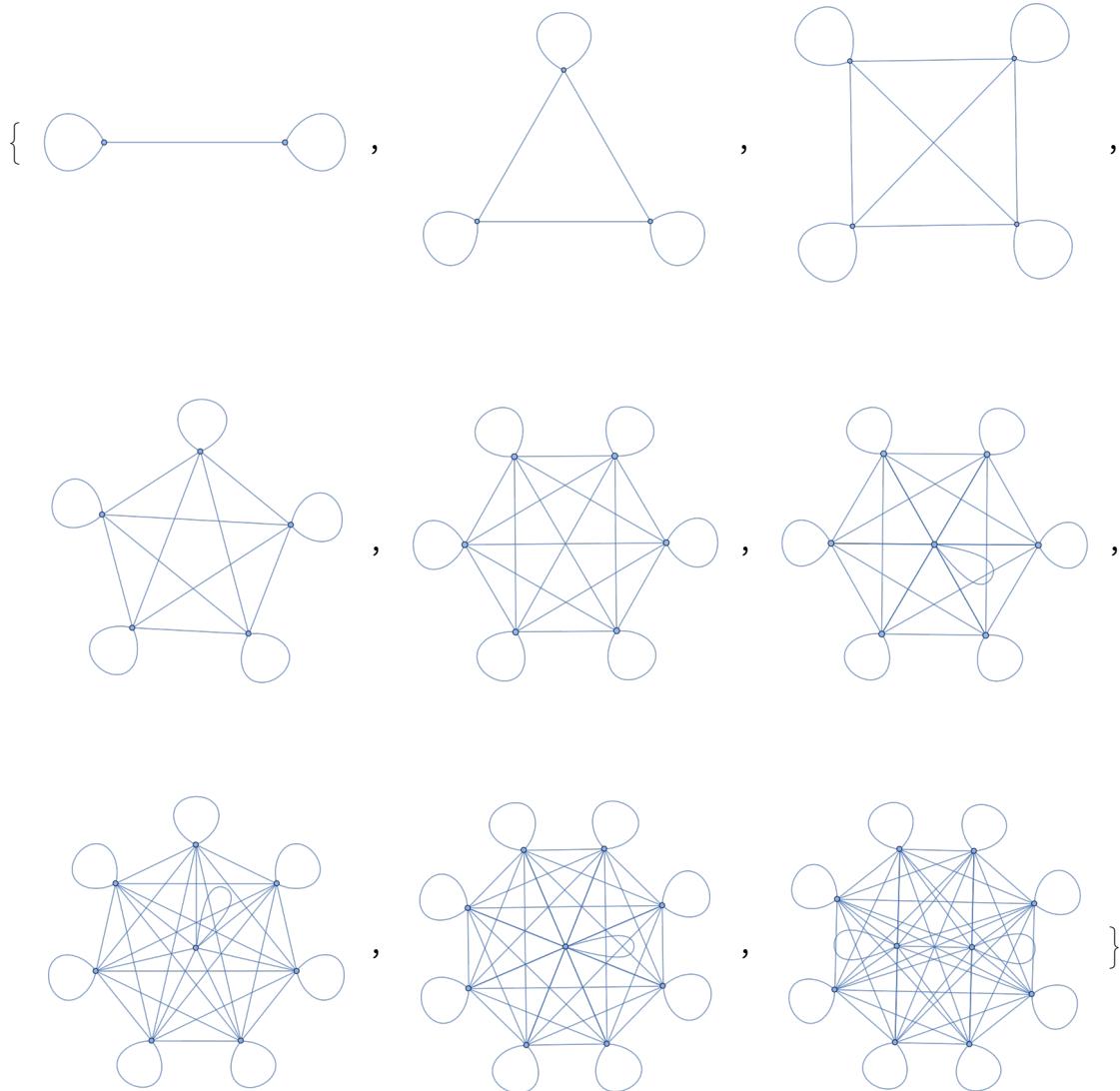
```
Out[101]=
```



In[102]:=

```
Table[UndirectedGraph[Flatten[Table[i → j, {i, n}, {j, n}]]], {n, 2, 10}]
```

Out[102]=



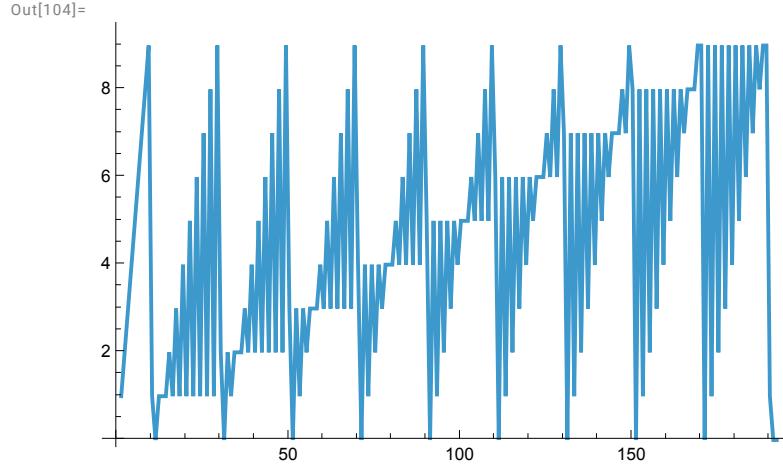
In[103]:=

```
Flatten[Table[{1, 2}, 3]]
```

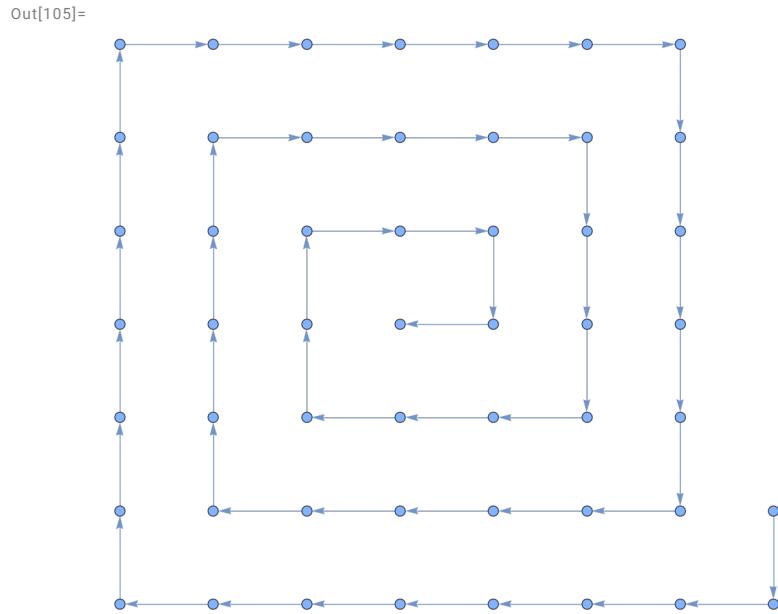
Out[103]=

```
{1, 2, 1, 2, 1, 2}
```

In[104]:= `ListLinePlot[Flatten[IntegerDigits[Range[100]]]]`



In[105]:= `Graph[Flatten[Table[i \rightarrow i + 1, {i, 50}]]]`



In[106]:= `Table[i \rightarrow Max[i, j], j \rightarrow Max[i, j], {i, 4}, {j, 4}]`

••• **Table**: Non-list iterator `j \rightarrow Max[i, j]` at position 2 does not evaluate to a real numeric value.

Out[106]= `Table[i \rightarrow Max[i, j], j \rightarrow Max[i, j], {i, 4}, {j, 4}]`

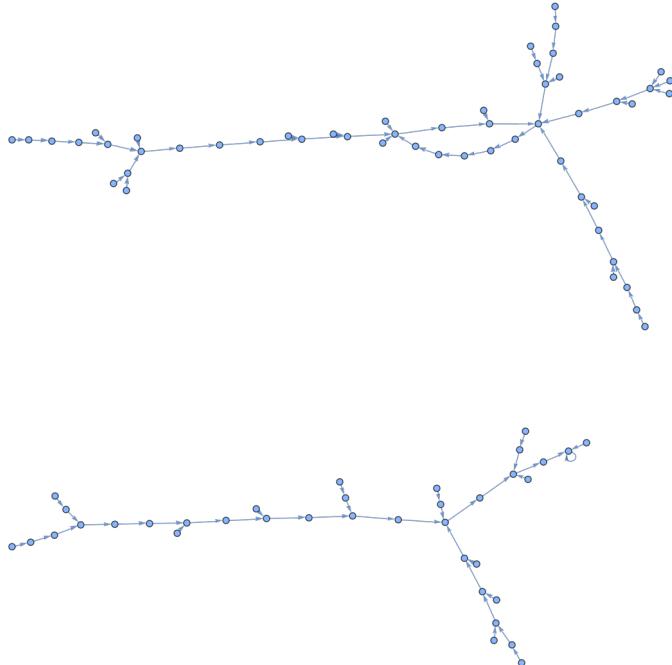
In[107]:= `Graph[Flatten[Table[i \rightarrow (j - i), j \rightarrow (j - i), {i, 1, 5}, {j, 1, 5}]]]`

••• **Table**: Non-list iterator `j \rightarrow j - i` at position 2 does not evaluate to a real numeric value.

Out[107]= `Graph[Table[i \rightarrow j - i, j \rightarrow j - i, {i, 1, 5}, {j, 1, 5}]]`

```
In[108]:= Graph[Table[i → RandomInteger[100], {i, 100}]]
```

```
Out[108]=
```



```
In[109]:=
```

```
Grid[Table[FindShortestPath[
  Graph[{1 → 2, 2 → 3, 3 → 4, 4 → 1, 3 → 1, 2 → 2}], a, b], {a, 4}, {b, 4}]]
```

```
Out[109]=
```

{1}	{1, 2}	{1, 2, 3}	{1, 2, 3, 4}
{2, 3, 1}	{2}	{2, 3}	{2, 3, 4}
{3, 1}	{3, 1, 2}	{3}	{3, 4}
{4, 1}	{4, 1, 2}	{4, 1, 2, 3}	{4}

Chapter 22

```
In[110]:=
```

```
LanguageIdentify["ajatella"]
```

```
Out[110]=
```

Finnish

```
ImageIdentify[tiger SPECIES SPECIFICATION   ["Image"]]
```

```
Out[111]=
```

tiger

```

Table[ImageIdentify[Blur[tiger SPECIES SPECIFICATION ... ✓ ["Image"], n]], {n, 5}]

Out[112]=
{tiger, tiger, tiger, tiger, swift fox}

In[113]:= Classify["Sentiment", "I'm so happy to be here"]
Out[113]=
Positive

In[114]:= Nearest[WordList[], "happy", 10]
Out[114]=
{happy, haply, harpy, nappy, sappy, apply, campy, choppy, guppy, hairy}

In[115]:= Nearest[Table[RandomInteger[1000], 20], 1000, 3]
Out[115]=
{998, 952, 951}

In[116]:= Nearest[Table[RandomColor[], 10], RGBColor[1, 0, 0], 5]
Out[116]=
{█, █, █, █, █}

In[117]:= Nearest[Table[n^2, {n, 100}], 2000]
Out[117]=
{2025}

Nearest[EntityValue[Europe GEOGRAPHIC REGION [countries], "Flag"],
Brazil COUNTRY ... ✓ ["Flag"], 3]

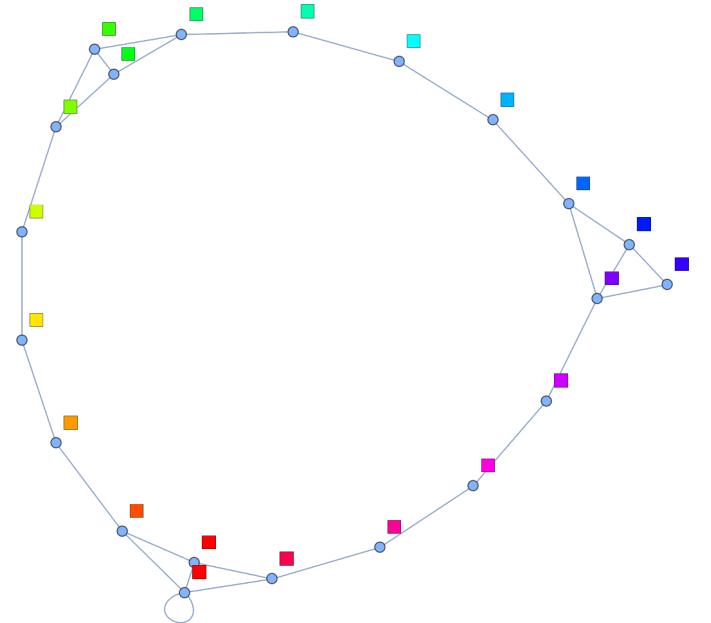
Out[118]=
{, , }

```

In[119]:=

```
NearestNeighborGraph[Table[Hue[h], {h, 0, 1, .05}], 2, VertexLabels → All]
```

Out[119]=



In[120]:=

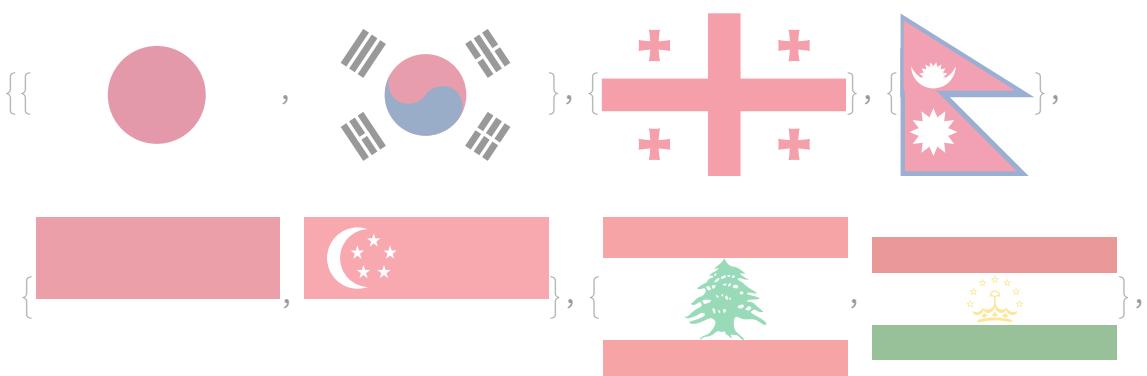
```
NearestNeighborGraph[Table[RandomInteger[100], 100], 2, VertexLabels → All]
```

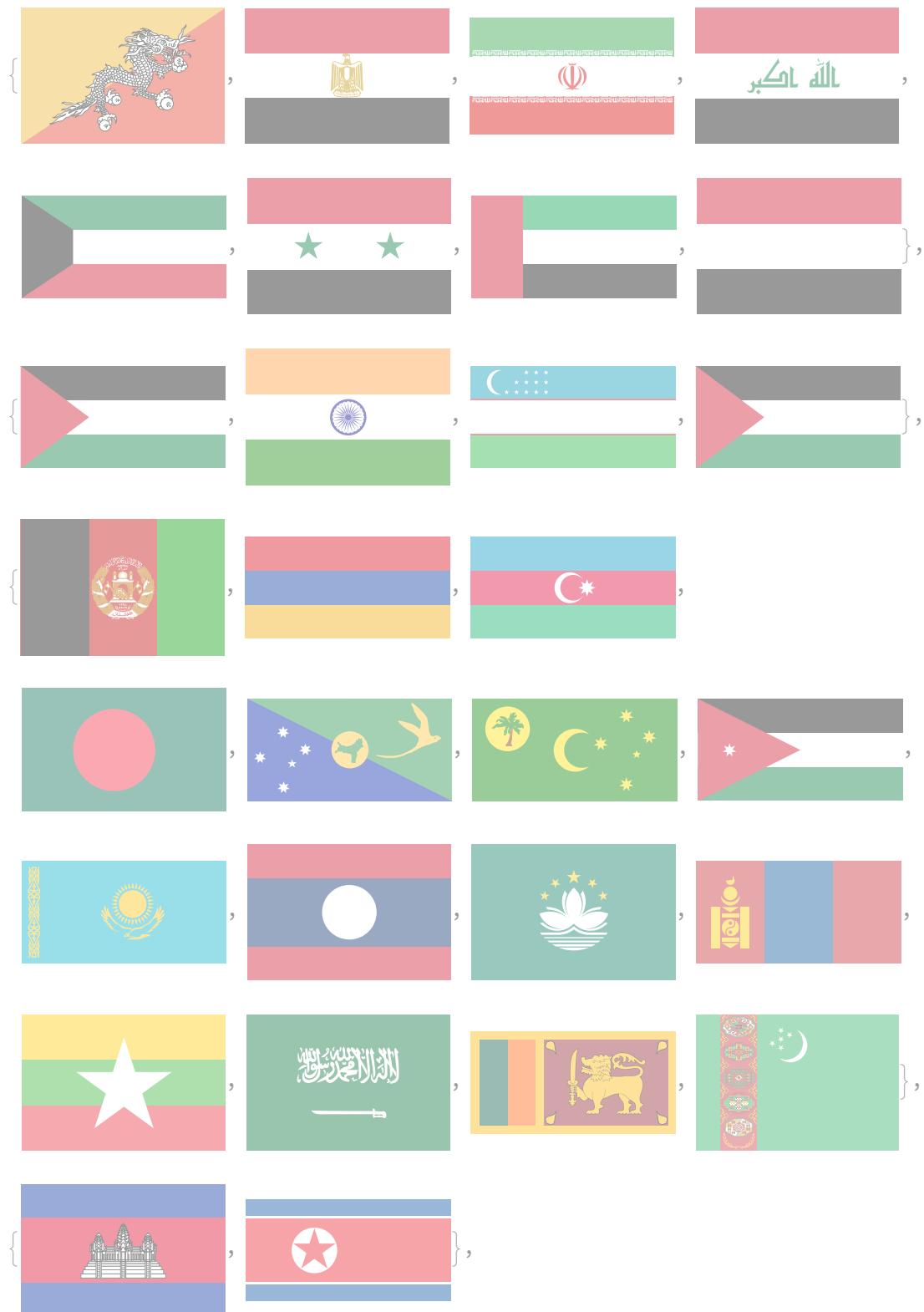
Out[120]=

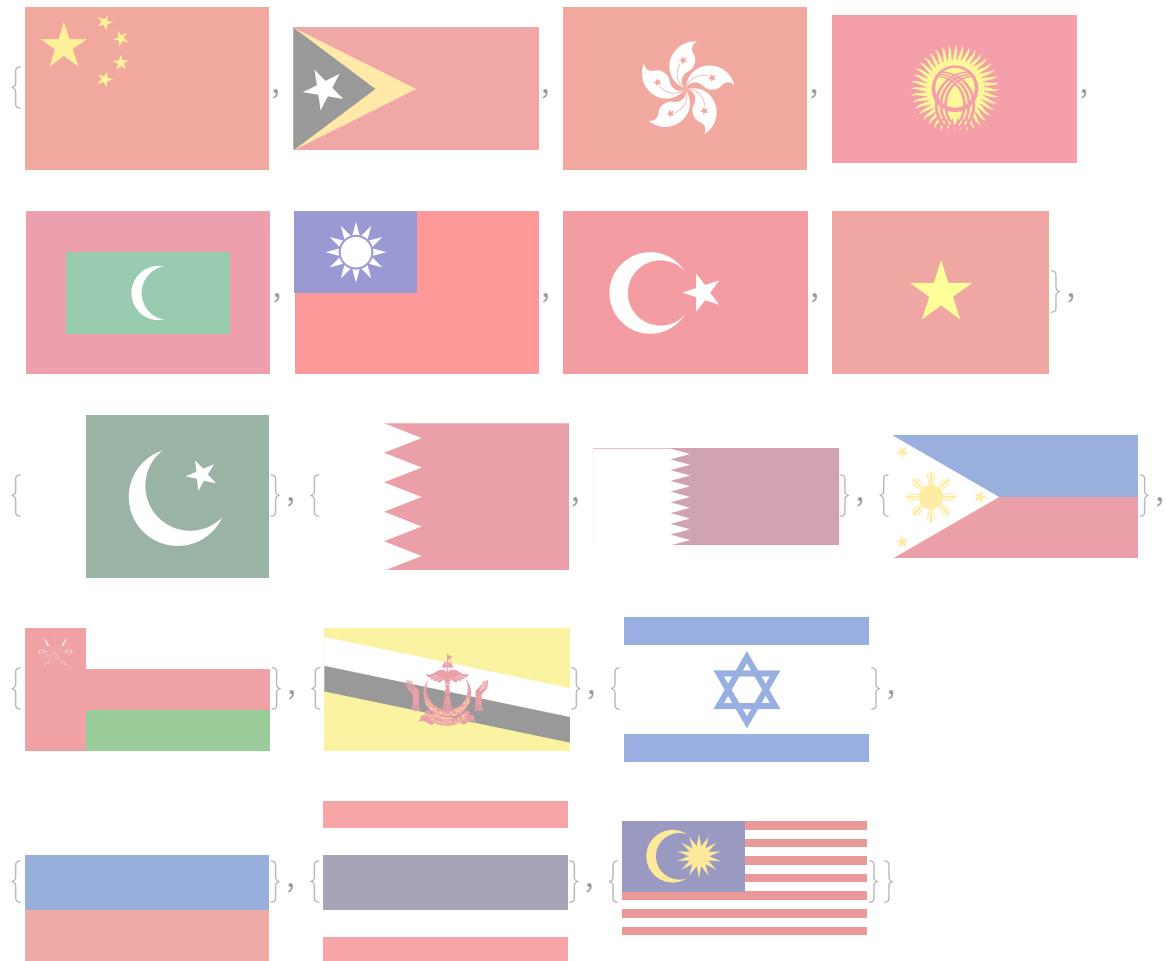


```
FindClusters[EntityValue[ , "Flag"]]
```

Out[121]=



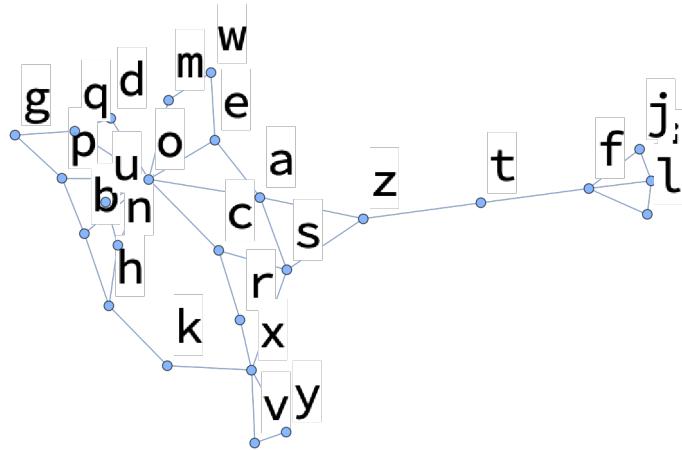




In[122]:=

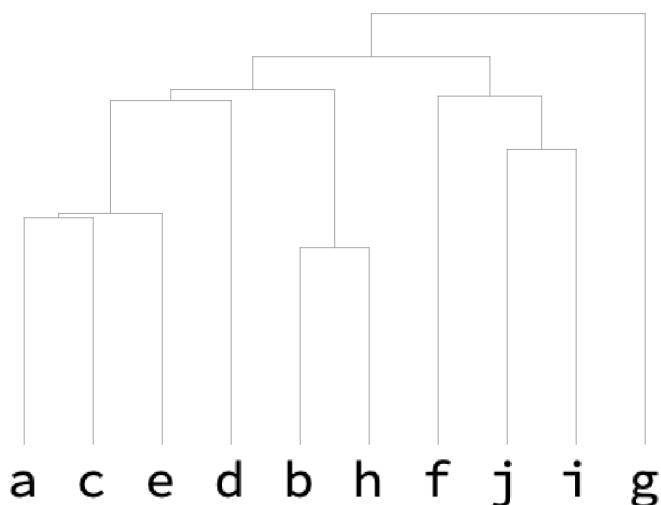
```
NearestNeighborGraph[
Table[Rasterize[Style[FromLetterNumber[n], 20]], {n, 26}], 2, VertexLabels -> All]
```

Out[122]=



```
In[123]:= Table[TextRecognize[EdgeDetect[Rasterize[Style["Programming ", n]]]], {n, 1, 10}]
Out[123]= {, , , , Programming, Programming, Programming, Programming, Programming}
```

```
In[124]:= Dendrogram[Table[Rasterize[FromLetterNumber[n]], {n, 10}]]
Out[124]=
```



```
In[125]:= FeatureSpacePlot[Capitalize[Alphabet[]]]
Out[125]=
```

