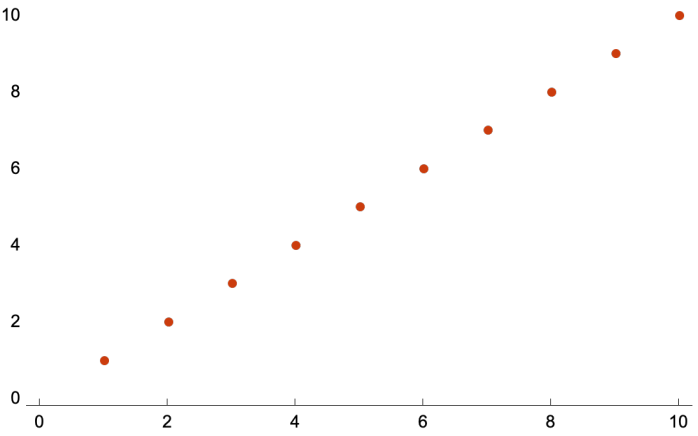
Brian — PS 8 — 2025-02-18 — Solution

EIWL3 Sections 20, 21, and 22

Exercises from *EIWL3* Section 20

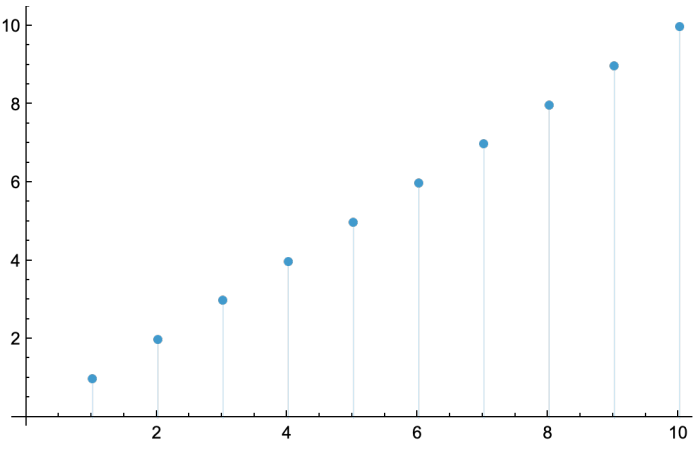
In[ ]:=
```
(* 20.1 *) ListPlot[Range[10], PlotTheme → "Web"]
```

Out[ ]=



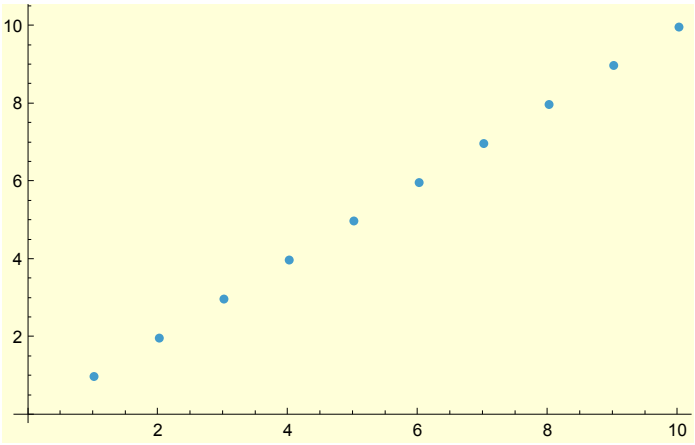
In[ ]:=
```
(* 20.2 *) ListPlot[Range[10], Filling → Axis]
(* That is interesting filling. I wonder if *)
(* Wolfram wanted a list line plot. *)
```

Out[ ]=



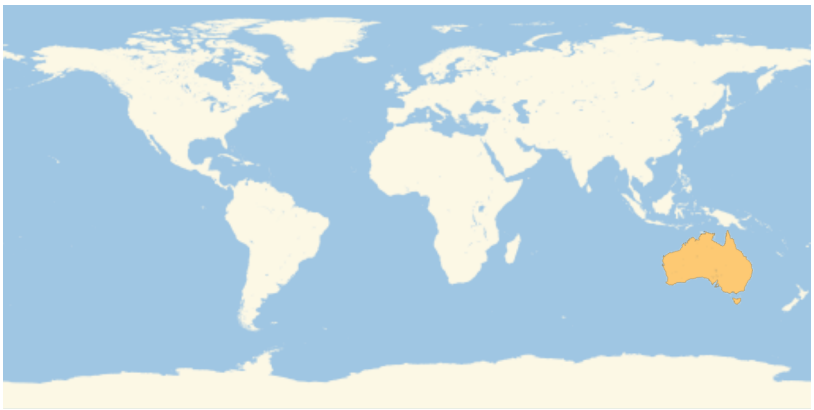
In[ ]:=
```
(* 20.3 *) ListPlot[Range[10], Background → LightYellow]
(* I couldn't bear the bright yellow so I dimmed it. *)
```

Out[ ]=



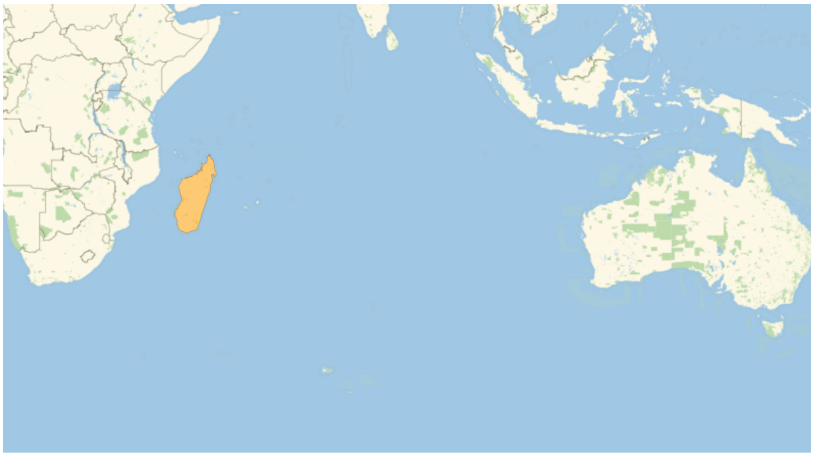
In[ ]:=
```
(* 20.4 *) GeoListPlot[ Australia COUNTRY , GeoRange → All]
```

Out[ ]=



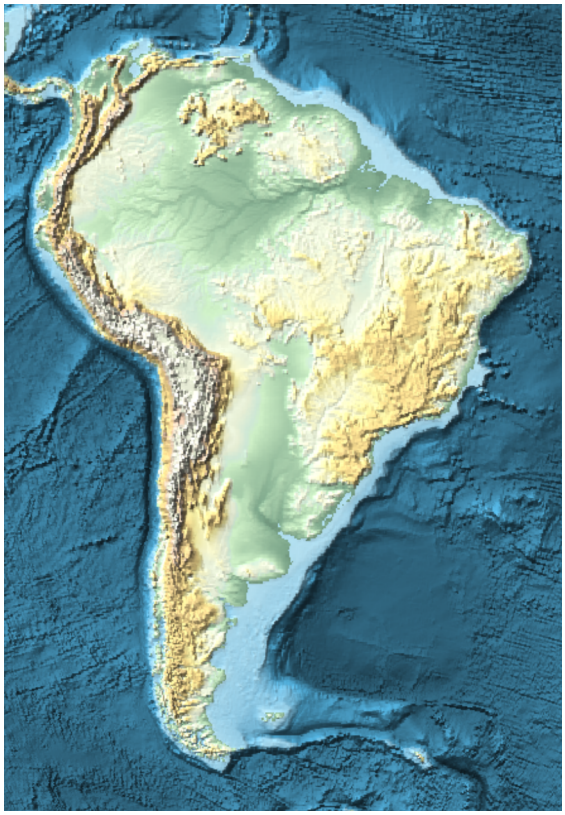
In[ ]:=
```
(* 20.5 *) GeoListPlot[ Madagascar COUNTRY , GeoRange → Indian Ocean OCEAN ]
```

Out[ ]=



In[•]:=
```
(* 20.6 *) GeoGraphics[ South America COUNTRIES , GeoBackground → "ReliefMap"]
```

Out[•]=



In[•]:=
```
(* 20.7 *) GeoListPlot[{ France COUNTRY ✓ , Finland COUNTRY , Greece COUNTRY ✓ },
 GeoRange → Europe GEOGRAPHIC REGION , GeoLabels → True]
```

Out[•]=

In[•]:=
```
(* 20.8 *) GeoListPlot[ The Ivy League UNIVERSITIES , GeoLabels → True]
```

Out[•]=

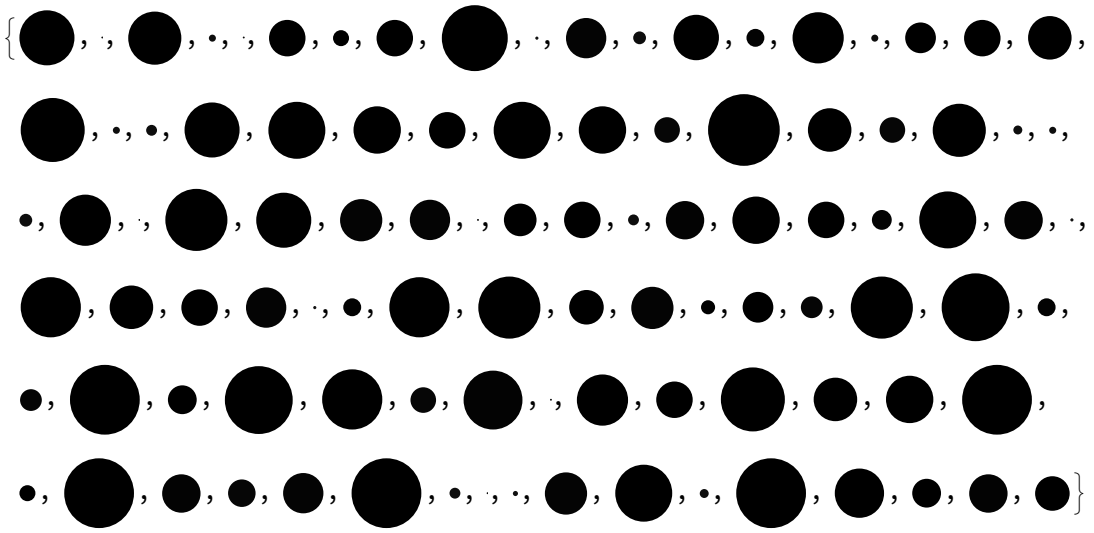
In[•]:=
```
(* 20.9 *)Style[Grid[ Table[x y, {x, 12}, {y, 12} ], Background → Black], White]
(* Is this the way we were meant to solve this? Well it works. *)
```

Out[•]=

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

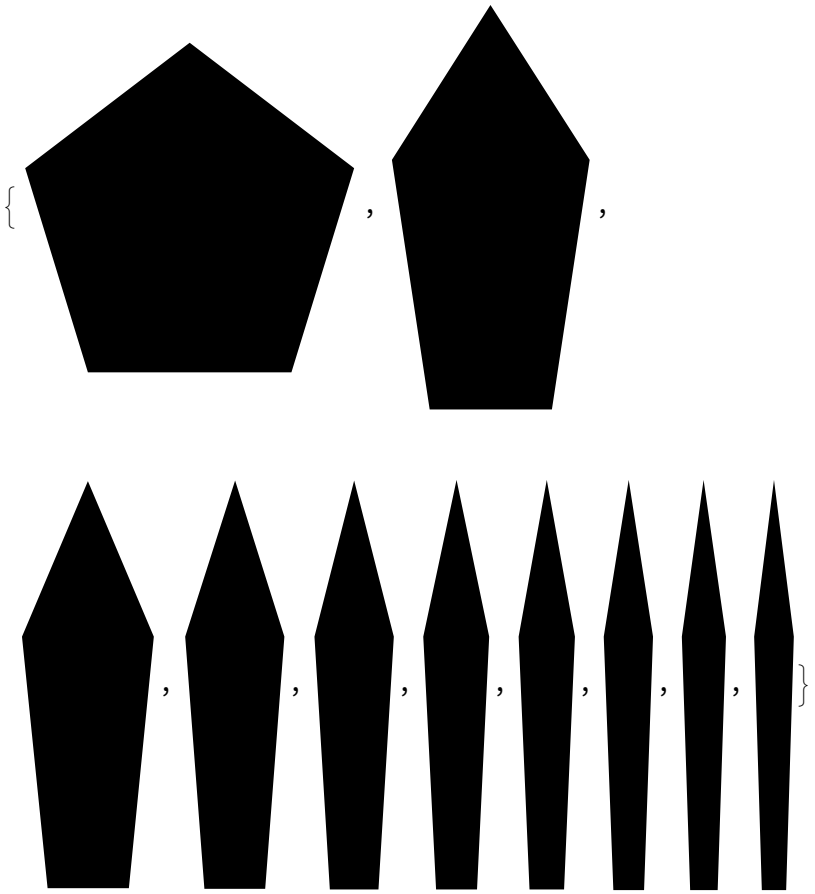
In[•]:=
```
(* 20.10 *) Table[Graphics[Disk[], ImageSize → RandomInteger[{1, 40}]], 100]
```

Out[•]=



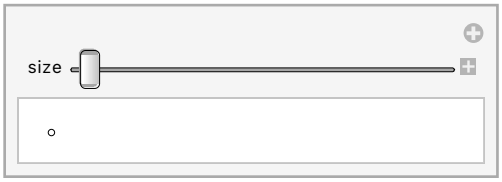
In[•]:=
```
(* 20.11 *)
Table[Graphics[RegularPolygon[5], AspectRatio → aspectRatio], {aspectRatio, 1, 10}]
```

Out[•]=



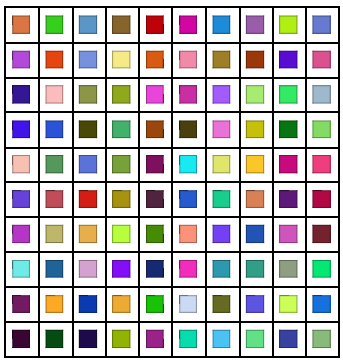
In[•]:=
```
(* 20.12 *) Manipulate[Graphics[Circle[], ImageSize → size], {size, 5, 500}]
```

Out[•]=



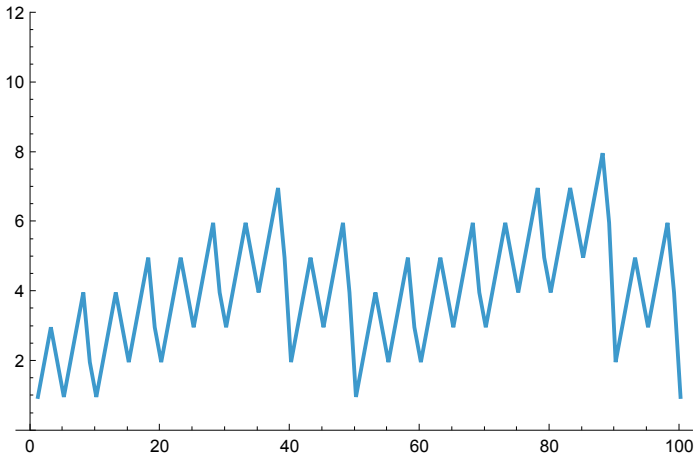
In[•]:=
```
(* 20.13 *) Grid[Table[RandomColor[], {i, 10}, {j, 10}], Frame → All]
```

Out[•]=



In[•]:=
```
(* 20.24 *) ListLinePlot[
 Table[StringLength[RomanNumeral[i]], {i, 100}],
 PlotRange → Max[Table[StringLength[RomanNumeral[i]], {i, 1000}]]]
```

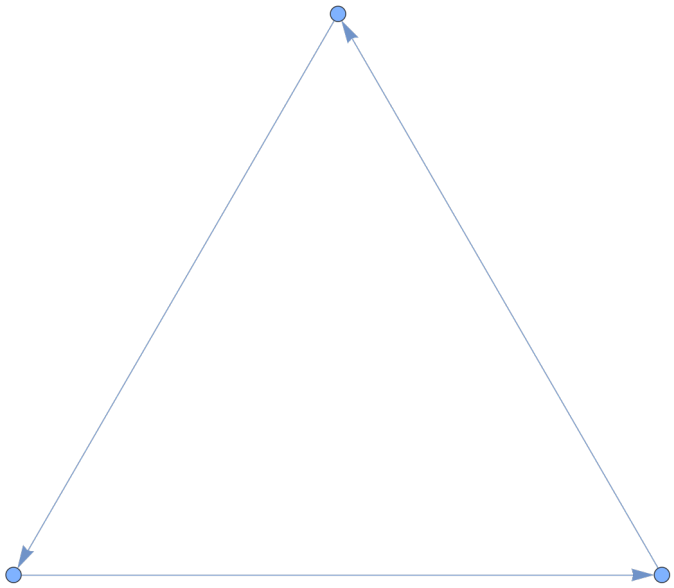
Out[•]=



Exercises from *EIWL3* Section 21

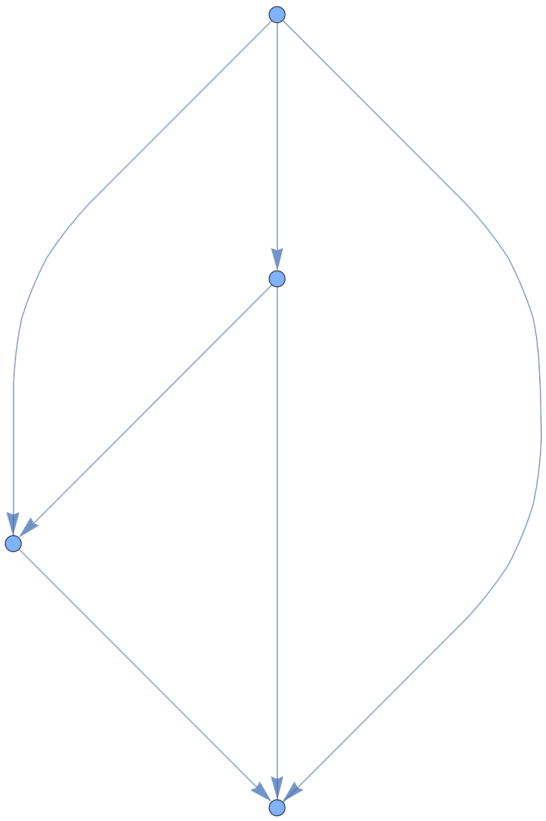
In[•]:=
```
(* 21.1 *) Graph[{1 → 2, 2 → 3, 3 → 1}]
```

Out[•]=



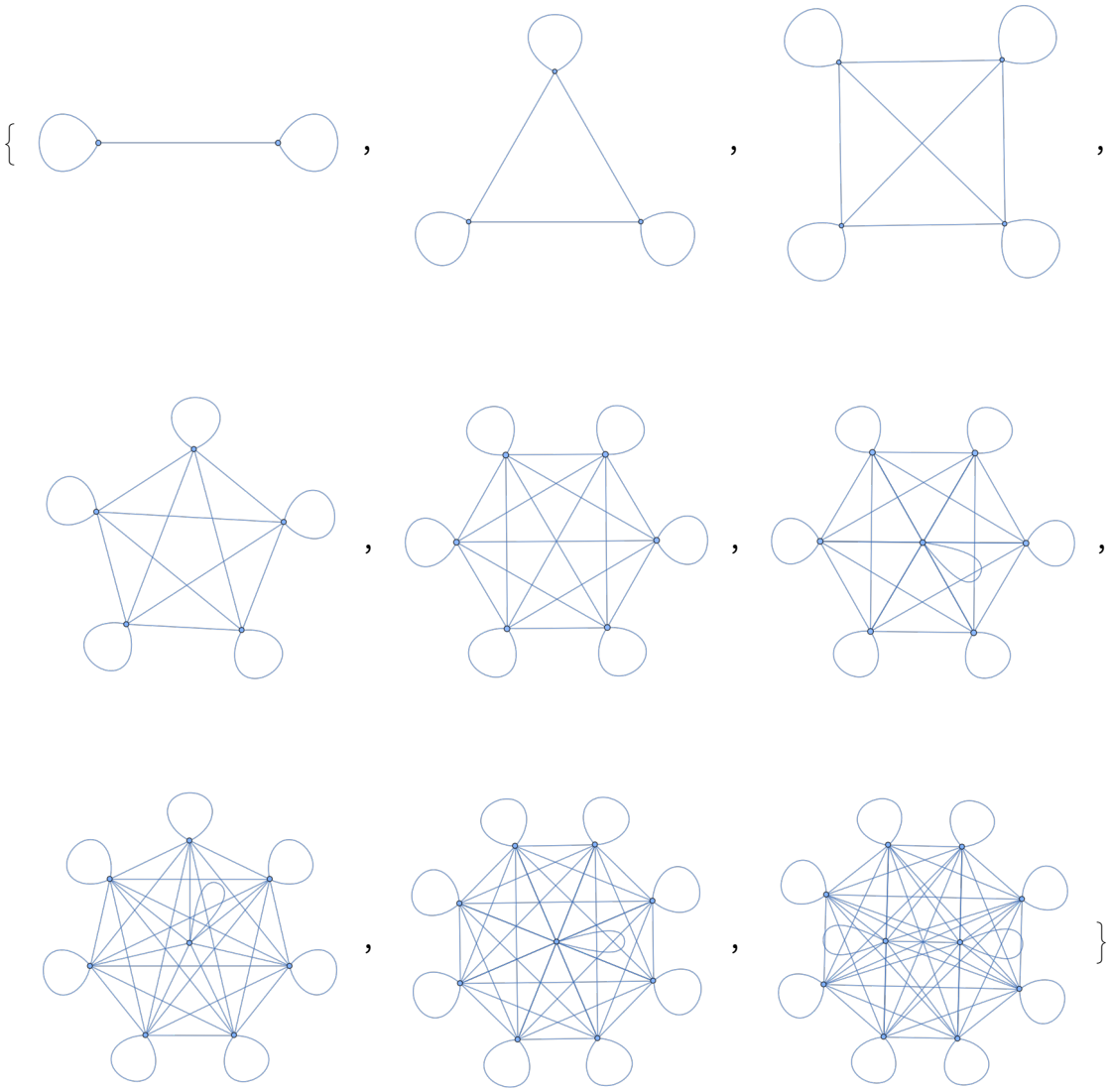
In[•]:=
```
(* 21.2 *) Graph[{1 → 2, 1 → 3, 1 → 4, 2 → 3, 2 → 4, 3 → 4}]
```

Out[•]=



In[ ]:=
```
(* 21.3 *)
Table[UndirectedGraph[Flatten[Table[i → j, {i, nodes}, {j, nodes}]]], {nodes, 2, 10}]
```

Out[ ]=

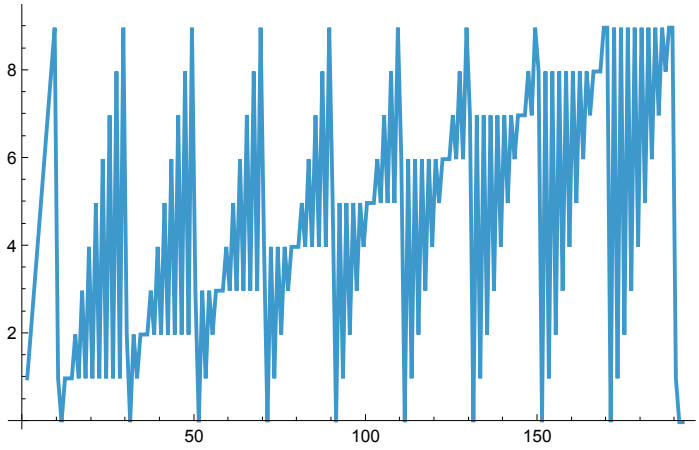


In[ ]:=
```
(* 21.4 *) Flatten[Table[i, 3, {i, 2}]]
```

Out[ ]=
```
{1, 2, 1, 2, 1, 2}
```

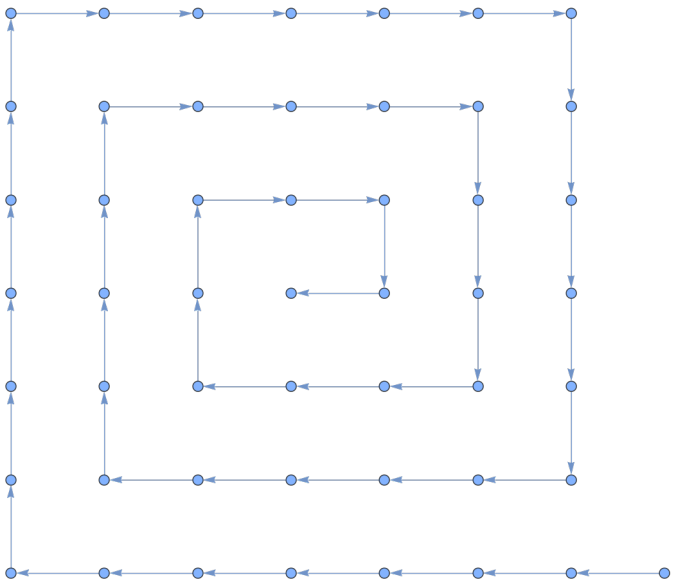
In[ ]:=
```
(* 21.5 *) ListLinePlot[Flatten[Table[IntegerDigits[number], {number, 100}]]]
```

Out[ ]=



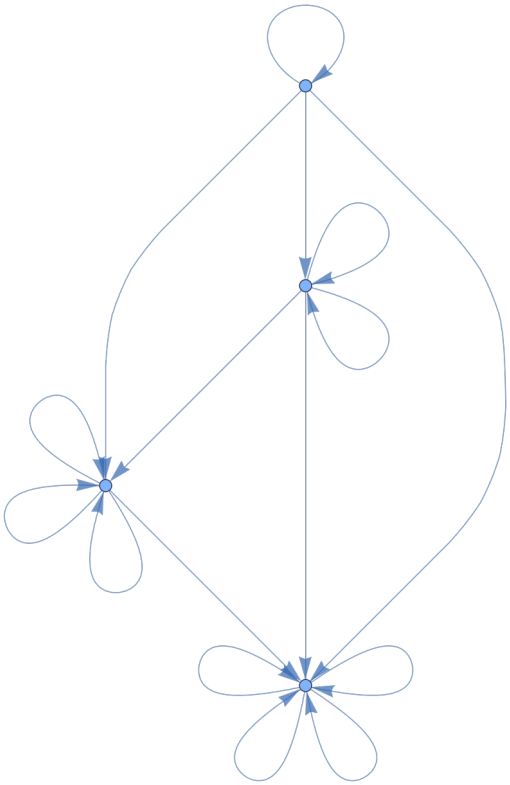
In[ ]:=
```
(* 21.6 *) Graph[Table[i → i + 1, {i, 1, 49}]]
```

Out[ ]=



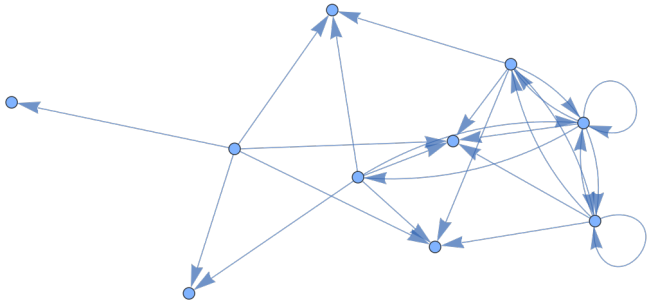
In[•]:=
```
(* 21.7 *) Graph[Flatten[Table[i → Max[i, j], {i, 4}, {j, 4}]]]
```

Out[•]=



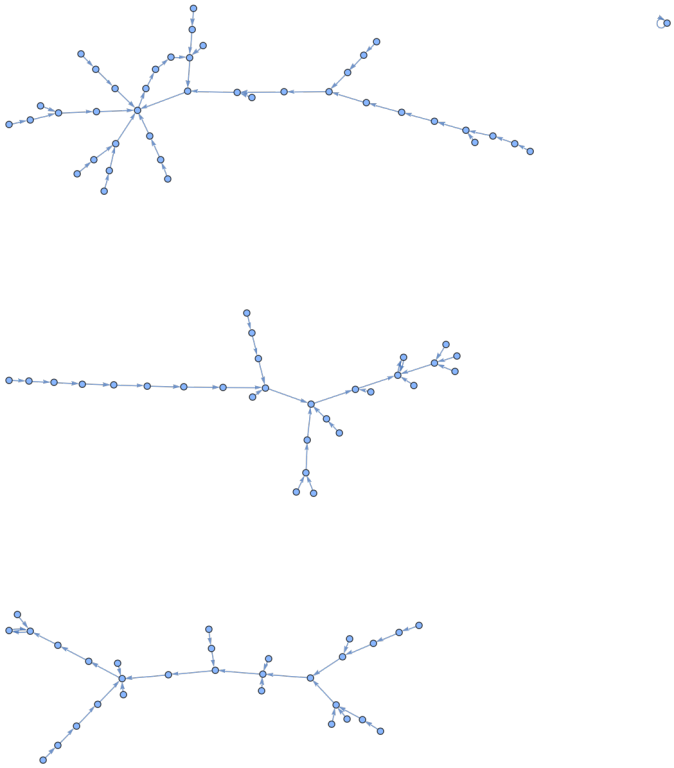
In[•]:=
```
(* 21.8 *) Graph[Flatten[Table[i → j - i, {i, 5}, {j, 5}]]]
```

Out[•]=



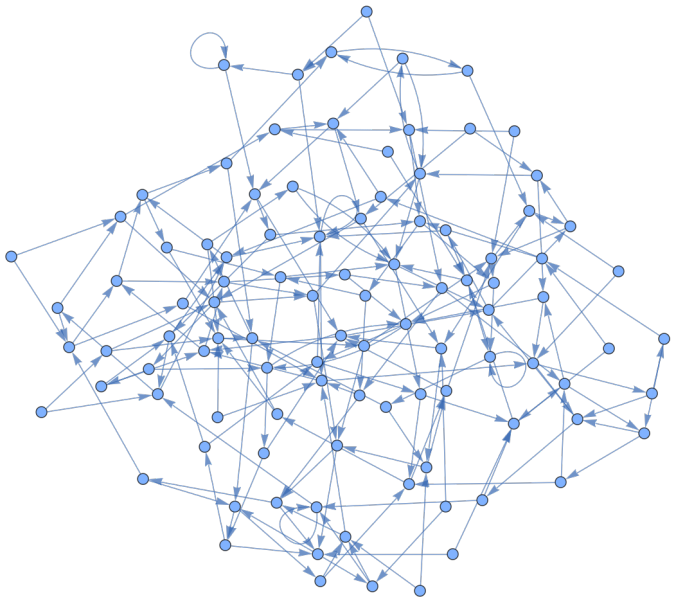
In[ ]:=
```
(* 21.9 *) Graph[Table[i → RandomInteger[{1, 100}], {i, 100}]]
```

Out[ ]=



In[ ]:=
```
(* 21.10 *) Graph[
 Flatten[Table[{i → RandomInteger[{1, 100}], i → RandomInteger[{1, 100}]}, {i, 100}]]]
```

Out[ ]=



```
In[•]:= (* 21.11 *) Grid[Table[FindShortestPath[
          Graph[{1 → 2, 2 → 3, 3 → 4, 4 → 1, 3 → 1, 2 → 2}], i, j], {i, 4}, {j, 4}]]
```

Out[•]=

{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}

Exercises from *EIWL3* Section 22

```
In[•]:= (* 22.1 *) LanguageIdentify["ajatella"]
```

Out[•]=

Finnish

```
In[•]:= (* This will be useful in the next two exercises: *)
        tigerImage = tiger SPECIES SPECIFICATION [ image ];
```

```
In[•]:= (* 22.2 *) ImageIdentify[tigerImage]
```

Out[•]=

tiger

```
In[•]:= (* 22.3 *)
        ImageIdentify[Table[Blur[tigerImage, blur], {blur, 5}]]
```

Out[•]=

{ tiger , tiger , tiger , tiger , swift fox }

```
In[•]:= (* 22.4 *) Classify["Sentiment", "I'm so happy to be here"]
```

Out[•]=

Positive

```
In[•]:= (* 22.5 *) Nearest[WordList[], "happy", 10]
```

Out[•]=

{happy, haply, harpy, nappy, sappy, apply, campy, choppy, guppy, hairy}

```
In[•]:= (* 22.6 *) Nearest[RandomInteger[1000, 20], 100, 3]
```

Out[•]=

{132, 6, 203}

```
In[•]:= (* 22.7 *) Nearest[RandomColor[10], Red, 5]
```

Out[•]=

{ , , , , }

```
In[•]:= (* 22.8 *) Nearest[Range[100]^2, 2000]
```

Out[•]=

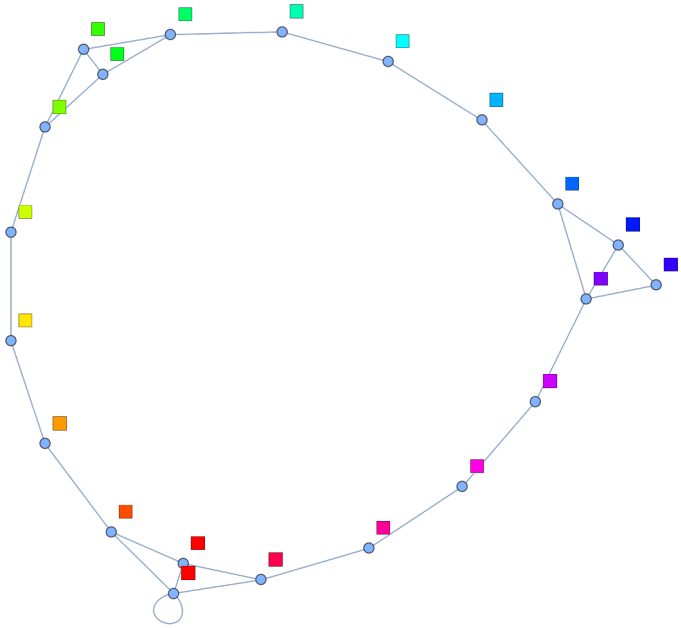
{2025}

In[•]:= (* 22.9 *) Nearest[Europe COUNTRIES [flag], Brazil COUNTRY [flag], 3]

Out[•]=

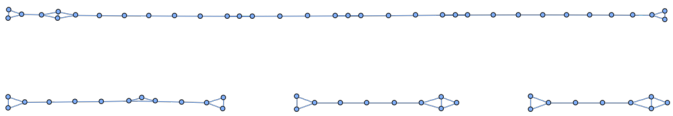
In[43]:= (* 22.10 *)
NearestNeighborGraph[Table[Hue[h], {h, 0, 1, 0.05}], 2, VertexLabels → All]

Out[43]=



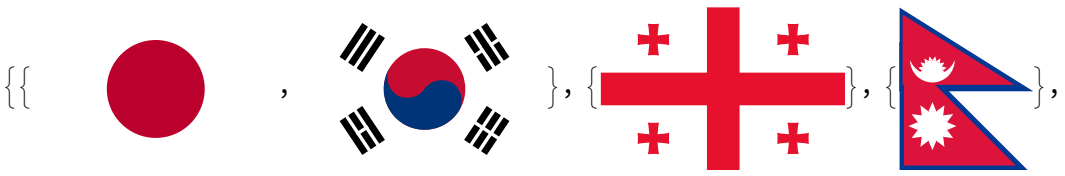
In[•]:= (* 22.11 *) NearestNeighborGraph[RandomInteger[100, 100], 2]

Out[•]=



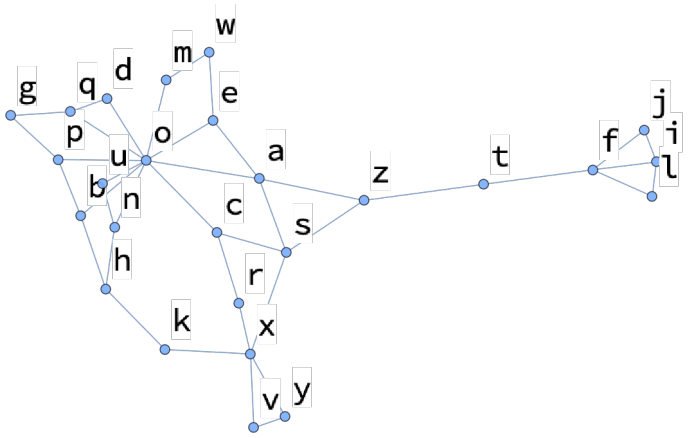
In[•]:= (* 22.12 *) FindClusters[Asia COUNTRIES [flag]]

Out[•]=



```
(* 22.13 *) NearestNeighborGraph[
 Table[Rasterize[letter, RasterSize → 20], {letter, Alphabet[]}],
 2, VertexLabels → All]
```

Out[44]=

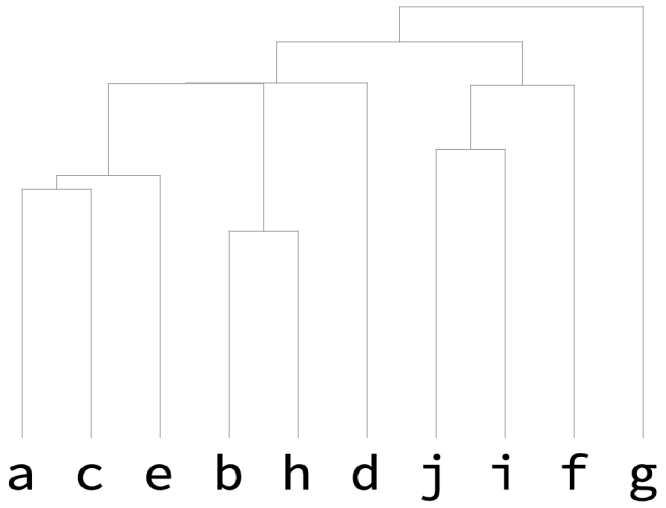


In[48]:=
```
(* 22.14 *) Table[
 TextRecognize[EdgeDetect[Rasterize[Style["programming", size]]]], {size, 10, 20}]
```

Out[48]=
```
{orogramming, programming, programming, programming, programming,
 programming, programming, programming, programming, programming, programming}
```

```
(* 22.15 *) Dendrogram[Table[Rasterize[Style[FromLetterNumber[n], 30]], {n, 10}]]
```

Out[63]=



In[55]:= `(* 22.16 *) FeatureSpacePlot[Rasterize /@ ToUpperCase[Alphabet[]]]`

Out[55]=