

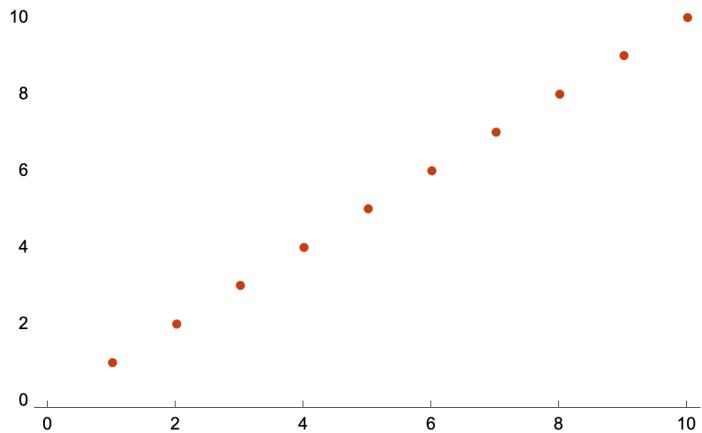
Very nice! 10/10. On p. 16, you didn't get 22.14 to work. No big deal.

Hexi—PS7—2025-02-10

Exercises from EIWL3 Section 20

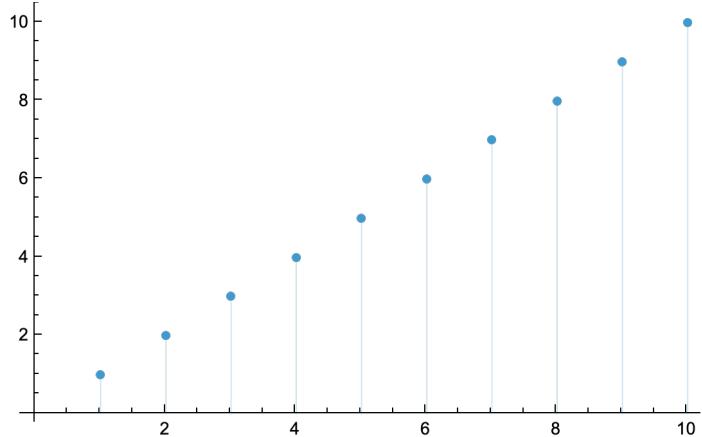
```
In[276]:= ListPlot[Range[10], PlotTheme -> "Web"]
```

```
Out[276]=
```



```
In[277]:= ListPlot[Range[10], Filling -> Axis]
```

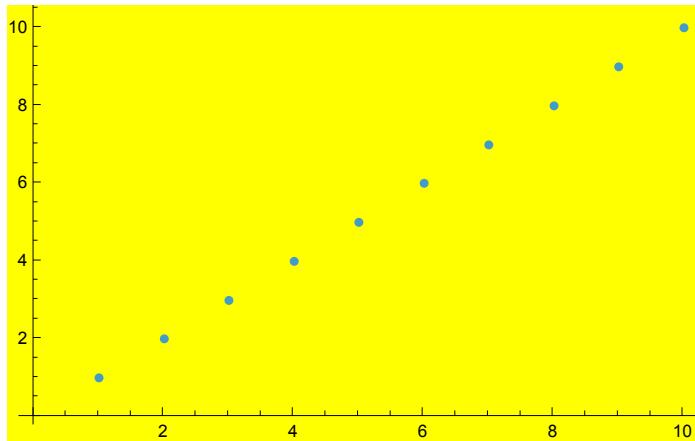
```
Out[277]=
```



In[278]:=

```
ListPlot[Range[10], Background -> Yellow]
```

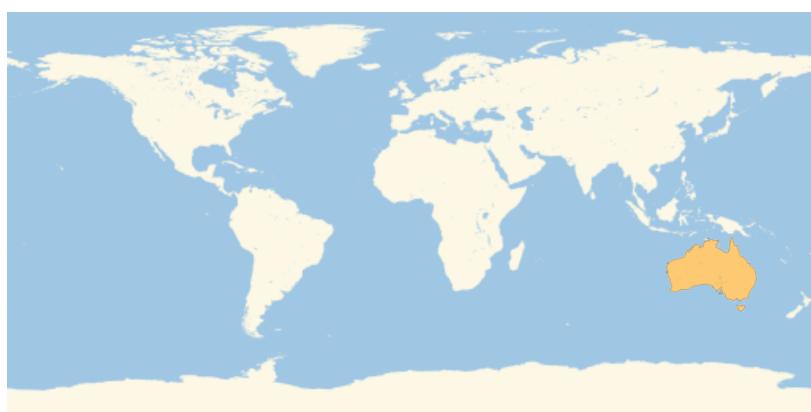
Out[278]=



In[279]:=

```
GeoListPlot[ Australia COUNTRY , GeoRange -> Earth PLANET ]
```

Out[279]=



In[280]:=

```
GeoListPlot[ Madagascar COUNTRY , GeoRange -> Indian Ocean OCEAN ]
```

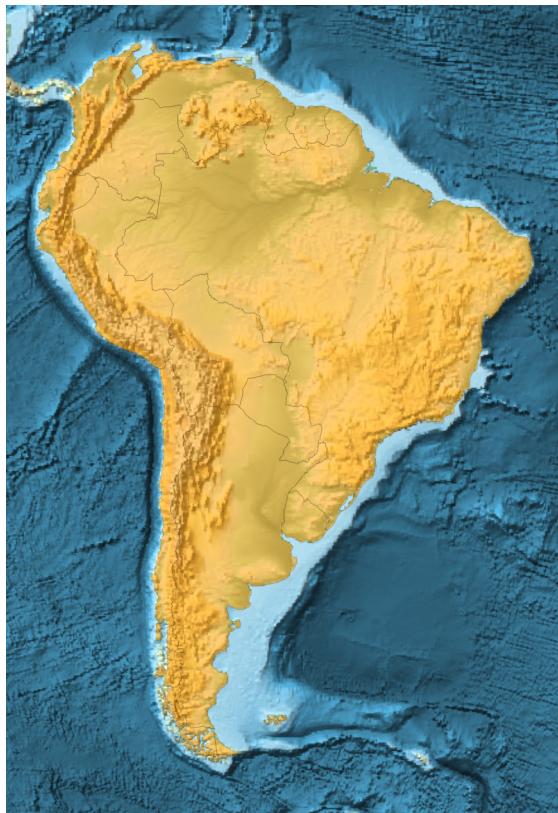
Out[280]=



In[281]:=

```
GeoListPlot[{"South America COUNTRIES"}, GeoBackground -> "ReliefMap"]
```

Out[281]=



In[282]:=

```
GeoListPlot[{France COUNTRY, Finland COUNTRY, Greece COUNTRY},  
GeoRange -> "Europe GEOGRAPHIC REGION", GeoLabels -> Automatic]
```

Out[282]=



In[283]:=

```
GeoListPlot[{"The Ivy League UNIVERSITIES"}, GeoLabels → True]
```

Out[283]=



In[284]:=

```
Grid[Table[i * j, {i, 12}, {j, 12}], ItemStyle → White, Background → Black]
```

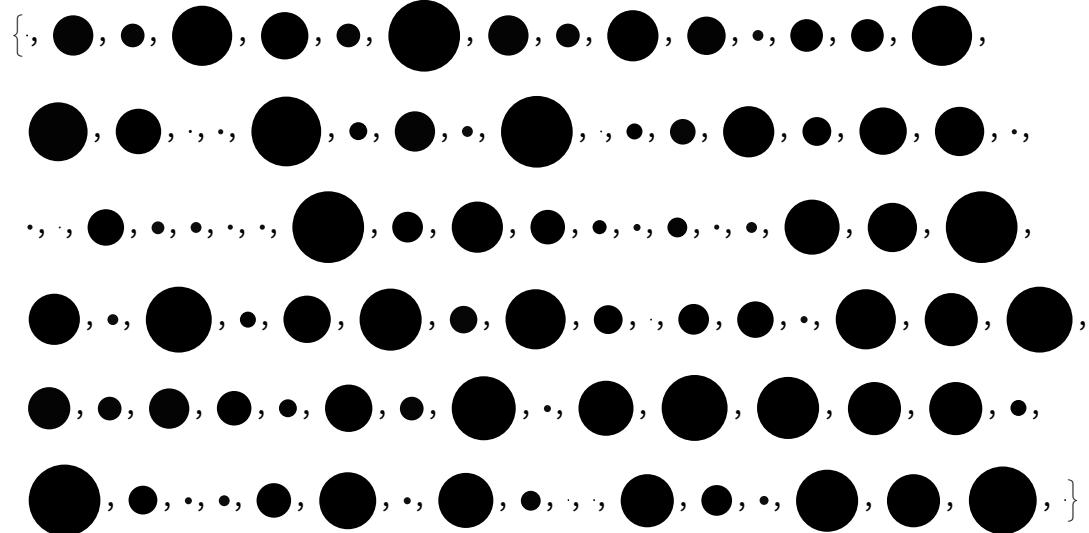
Out[284]=

| | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 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[285]:=

```
Table[Graphics[Disk[], ImageSize → RandomInteger[40]], 100]
```

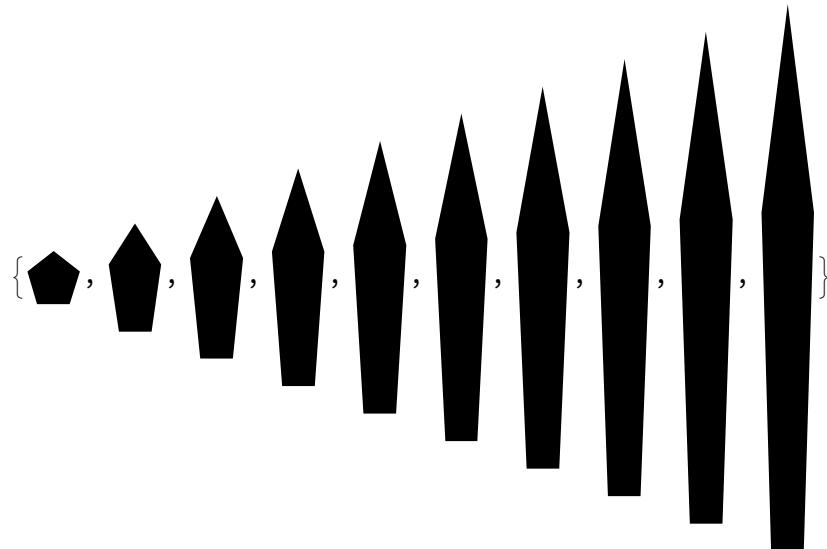
Out[285]=



In[286]:=

```
Table[Graphics[RegularPolygon[5], ImageSize → 30, AspectRatio → n], {n, 10}]
```

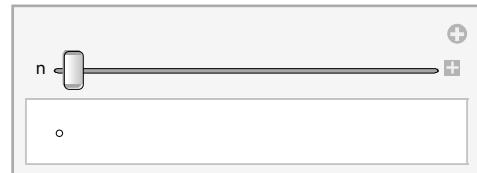
Out[286]=



In[287]:=

```
Manipulate[Graphics[Circle[], ImageSize → n], {n, 5, 500}]
```

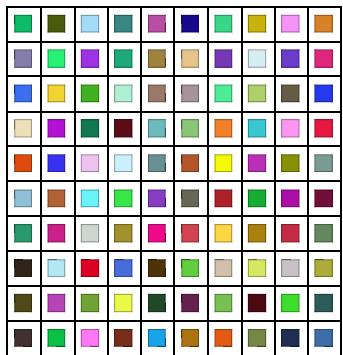
Out[287]=



In[288]:=

```
Grid[Table[RandomColor[], {10}, {10}], Frame -> All]
```

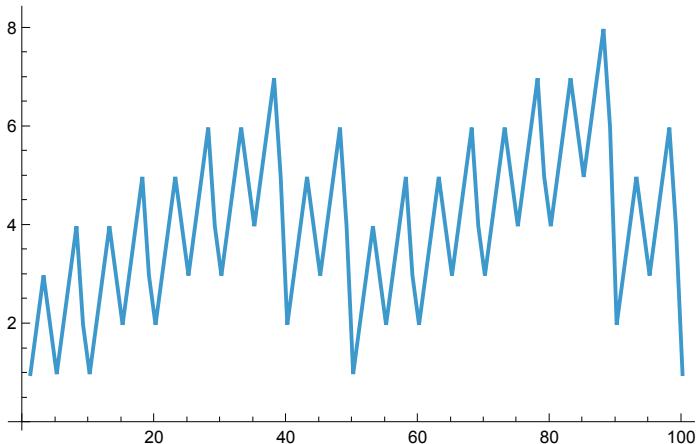
Out[288]=



In[289]:=

```
ListLinePlot[StringLength[RomanNumeral[Range[100]]], PlotRange -> All]
```

Out[289]=



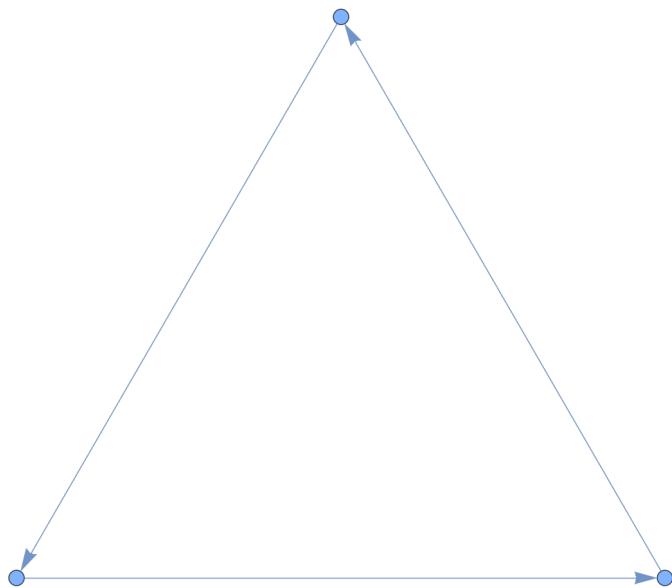
In[290]:=

Exercises from EIWL3 Section 21

In[291]:=

```
Graph[{1 → 2, 2 → 3, 3 → 1}]
```

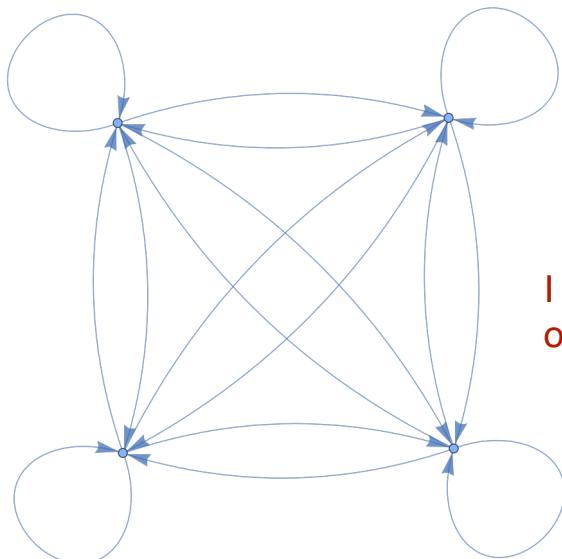
Out[291]=



In[292]:=

```
Graph[Flatten[Table[i → j, {i, 4}, {j, 4}]]]
```

Out[292]=

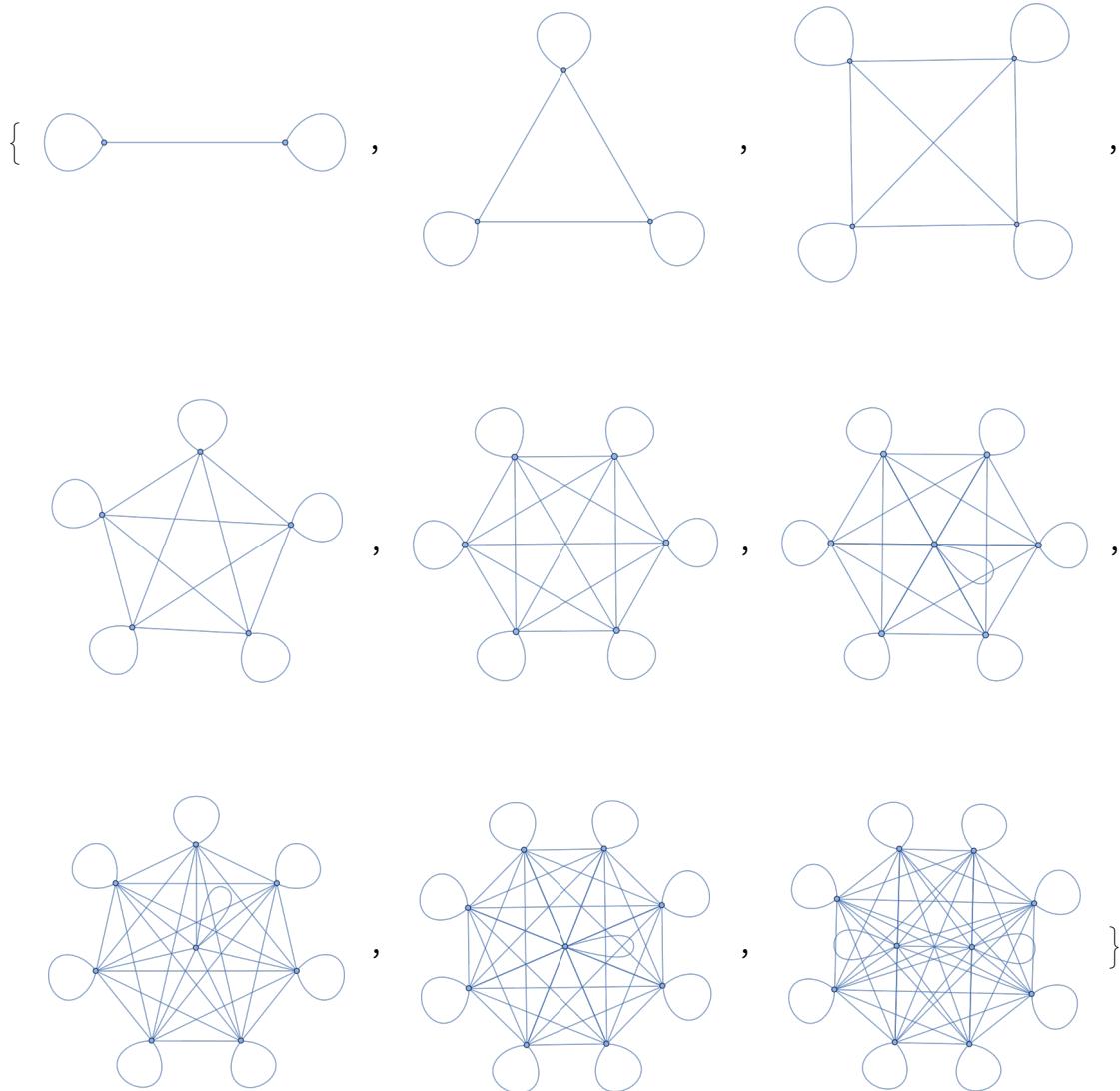


I had a different, simpler interpretation on this one.

In[293]:=

```
Table[UndirectedGraph[Flatten[Table[i → j, {i, n}, {j, n}]]], {n, 2, 10, 1}]
```

Out[293]=



In[294]:=

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

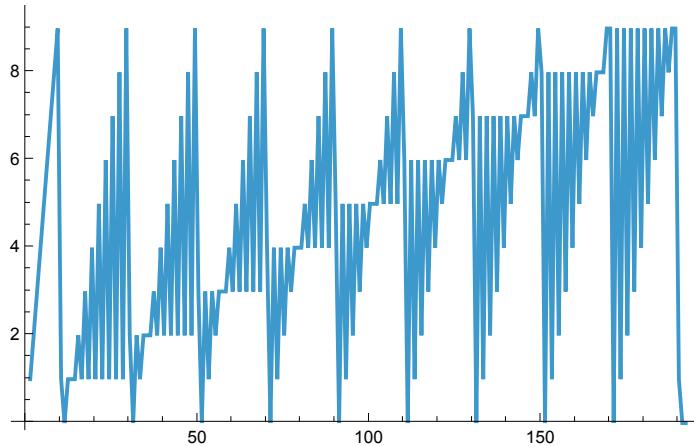
Out[294]=

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

In[295]:=

```
ListLinePlot[Flatten[IntegerDigits[Range[100]]]]
```

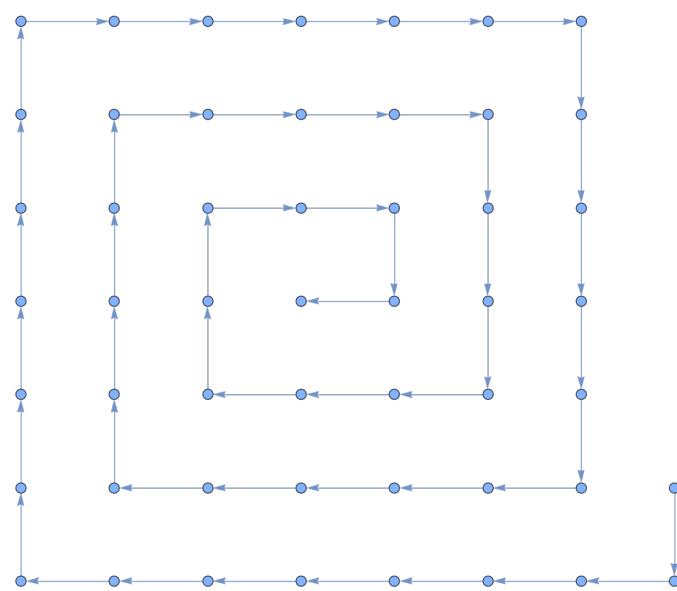
Out[295]=



In[296]:=

```
Graph[Table[i → i + 1, {i, 50}]]
```

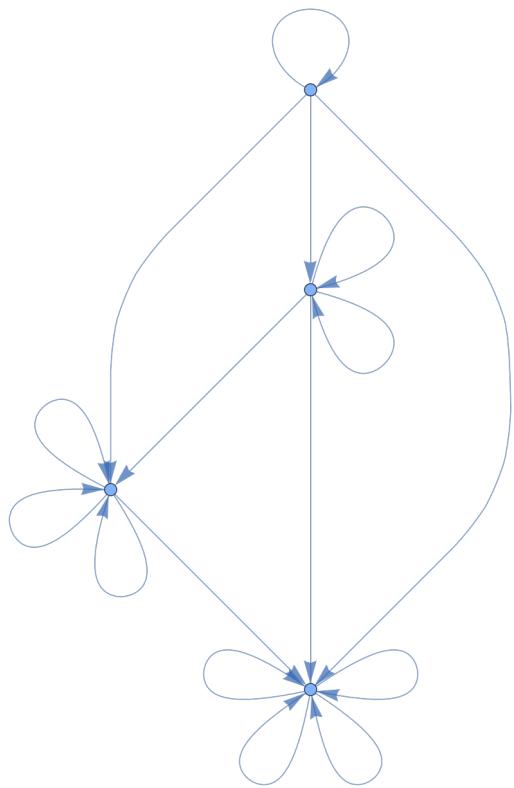
Out[296]=



In[297]:=

```
Graph[Flatten[Table[i → Max[i, j], {i, 1, 4}, {j, 1, 4}]]]
```

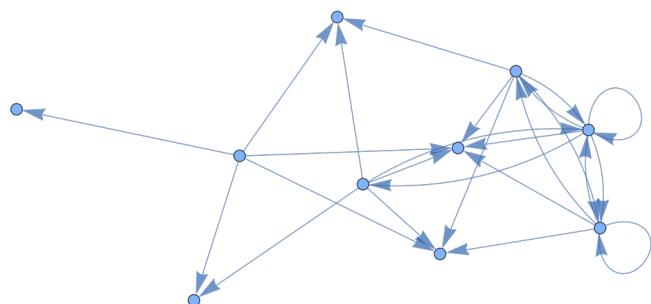
Out[297]=



In[298]:=

```
Graph[Flatten[Table[i → j - i, {i, 5}, {j, 5}]]]
```

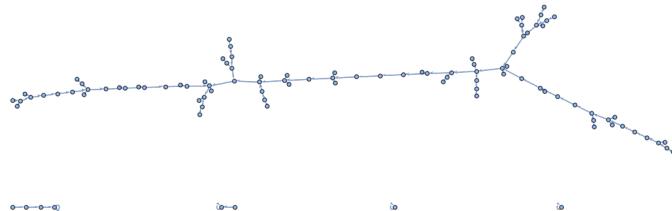
Out[298]=



In[299]:=

```
Graph[Flatten[Table[i → RandomInteger[100], {i, 100}]]]
```

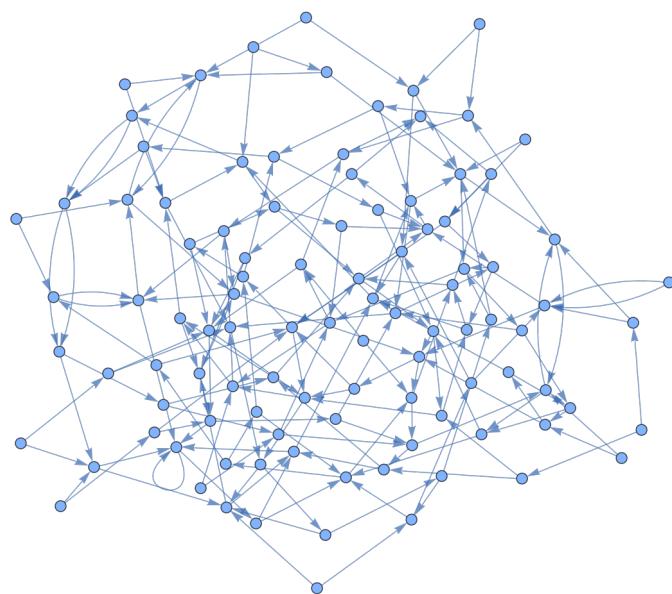
Out[299]=



In[300]:=

```
Graph[Flatten[Table[Table[i → RandomInteger[100], 2], {i, 100}]]]
```

Out[300]=



In[301]:=

```
Grid[
```

```
Table[FindShortestPath[{1 → 2, 2 → 3, 3 → 4, 4 → 1, 3 → 1, 2 → 2}, i, j], {i, 4}, {j, 4}]]
```

Out[301]=

| | | | |
|-----------|-----------|--------------|--------------|
| {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 20

In[302]:=

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

Out[302]=

Finnish

```
In[303]:= ImageIdentify[ tiger SPECIES SPECIFICATION ["Image"]]

Out[303]= tiger

In[304]:= Table[ImageIdentify[Blur[ tiger SPECIES SPECIFICATION ["Image"], r]], {r, 1, 5}]

Out[304]= { tiger , tiger , tiger , tiger , swift fox }

In[305]:= Classify["Sentiment", "I'm so happy to be here"]

Out[305]= Positive

In[306]:= Nearest[WordList[], "happy", 10]

Out[306]= {happy, haply, harpy, nappy, sappy, apply, campy, choppy, guppy, hairy}

In[307]:= Nearest[Table[RandomInteger[1000], 20], 100, 3]

Out[307]= {93, 130, 68}

In[308]:= Nearest[Table[RandomColor[], 10], Red, 5]

Out[308]= {■, ■, ■, ■, ■}

In[309]:= Nearest[ (Range[100])^2, 2000]

Out[309]= {2025}

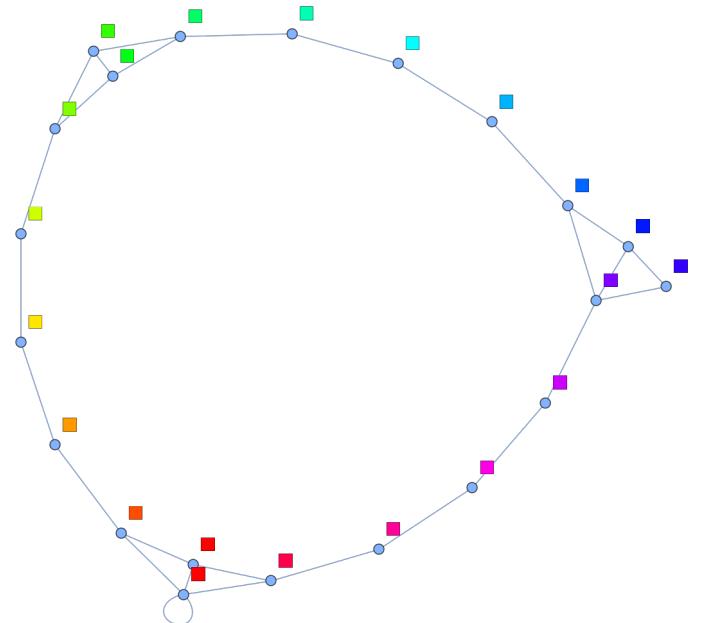
In[310]:= Nearest[ Europe COUNTRIES [ flag ], Brazil COUNTRY [ flag ], 3]

Out[310]= {  ,  ,  }
```

In[311]:=

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

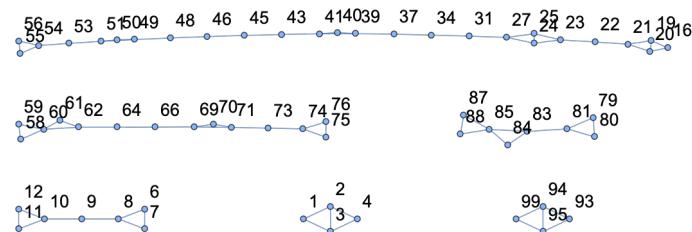
Out[311]=



In[312]:=

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

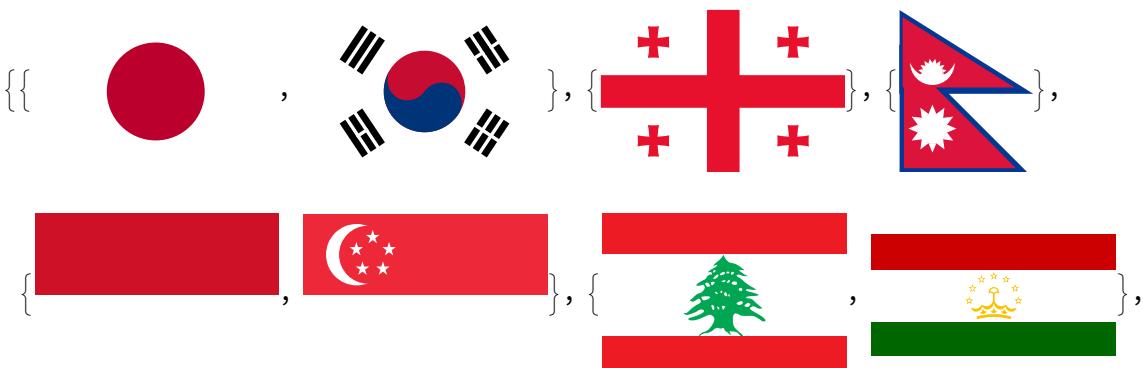
Out[312]=



In[313]:=

```
FindClusters[ Asia COUNTRIES [ flag]]
```

Out[313]=



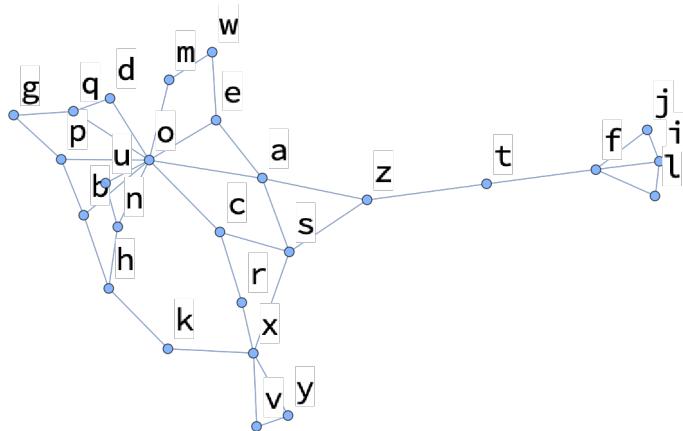




In[314]:=

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

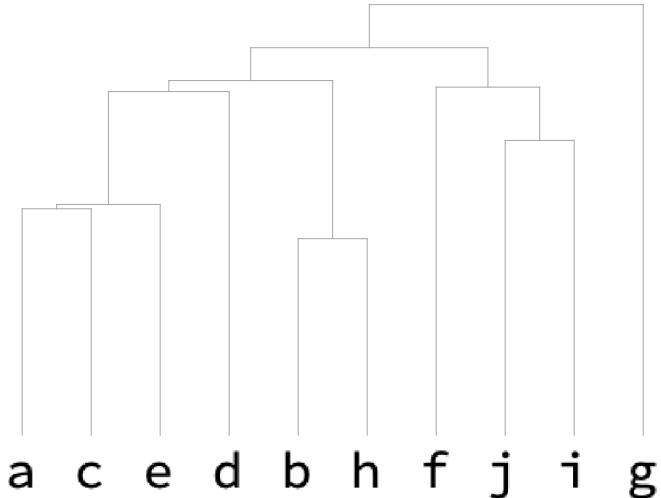
Out[314]=



```
In[315]:= Table[TextRecognize[EdgeDetect[Rasterize["programming", RasterSize -> n]]], {n, 10, 20, 1}]
Out[315]= {, , , , , , , , , }
```

See my solution to 22.14.

```
In[316]:= Dendrogram[Table[Rasterize[FromLetterNumber[n]], {n, 10}]]
Out[316]=
```



```
In[317]:= FeatureSpacePlot[Table[Rasterize[ToUpperCase[FromLetterNumber[n]]], {n, 26}]]
Out[317]=
```

