A few mistakes on pp. 2, 9, 10, bottom of 12, 13 and 15, but overall looks good. 9 1/2 / 10.

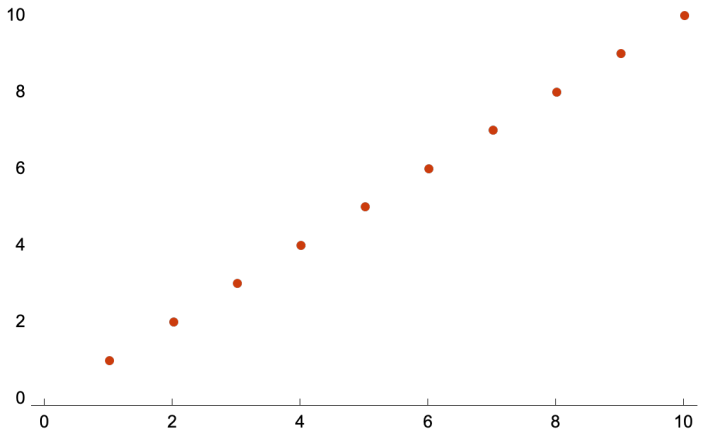
EIWL Sections 20-22

Section 20

In[235]:=

```
ListPlot[Range[10], PlotTheme → "Web"]
```

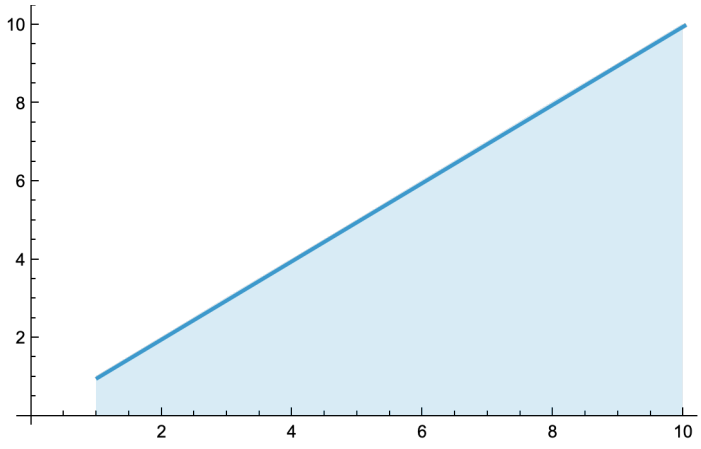
Out[235]=



In[236]:=

```
ListLinePlot[Range[10], Filling → Axis]
```

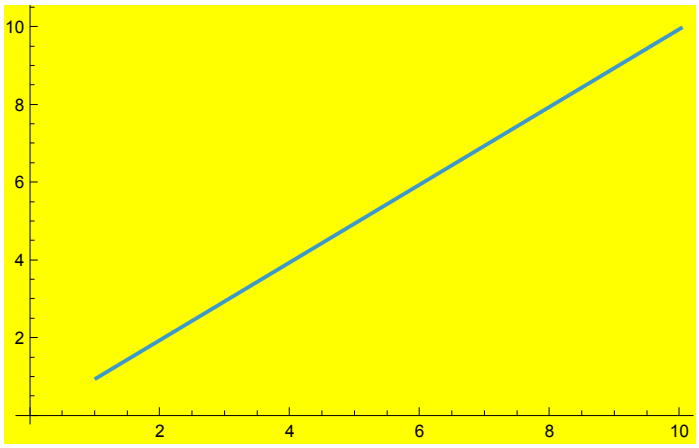
Out[236]=



I thought he wanted a ListPlot, but your ListLinePlot looks better.

In[237]:=
```
ListLinePlot[Range[10], Background → Yellow]
```

Out[237]=



Here too, I thought he wanted a ListPlot, not a ListLinePlot.

```
GeoListPlot[Australia COUNTRY, GeoRange → Europe GEOGRAPHIC REGION]
```

Out[238]=

Oops, Australia is not in Europe. He asked for the whole world as the GeoRange.

```
GeoListPlot[Madagascar COUNTRY ••• ✓, GeoRange → Indian Ocean OCEAN ••• ✓]
```

Out[239]=


```
GeoGraphics[South America COUNTRIES ••• ✓, GeoBackground → "ReliefMap"]
```

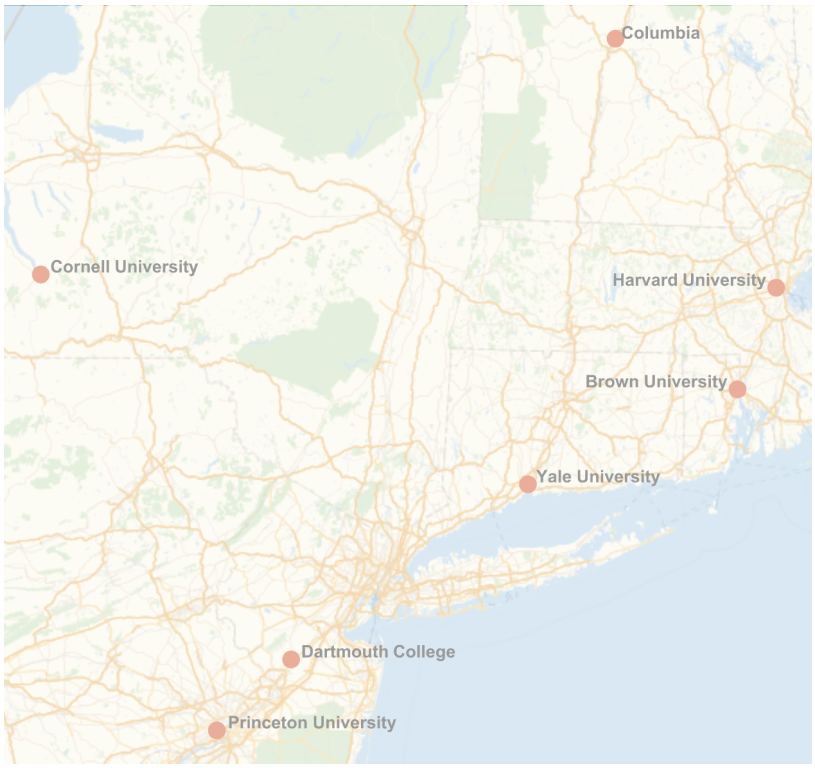
Out[240]=

```
GeoListPlot[{France COUNTRY ✓, Finland COUNTRY ••• ✓, Greece COUNTRY ✓},
 GeoRange → Europe GEOGRAPHIC REGION ••• ✓, GeoLabels → Automatic]
```

Out[241]=

```
GeoListPlot[{Yale University UNIVERSITY, Cornell University UNIVERSITY,
  Brown University UNIVERSITY, Harvard University UNIVERSITY,
  Columbia SPECIES SPECIFICATION, Dartmouth College UNIVERSITY,
  Princeton University UNIVERSITY, University of Pennsylvania UNIVERSITY}, GeoLabels → True]
```

Out[242]=



In[243]:=

```
ColorNegate[
 Rasterize[Grid[Table[x * y, {x, 12}, {y, 12}], Frame → All, Background → White]]]
```

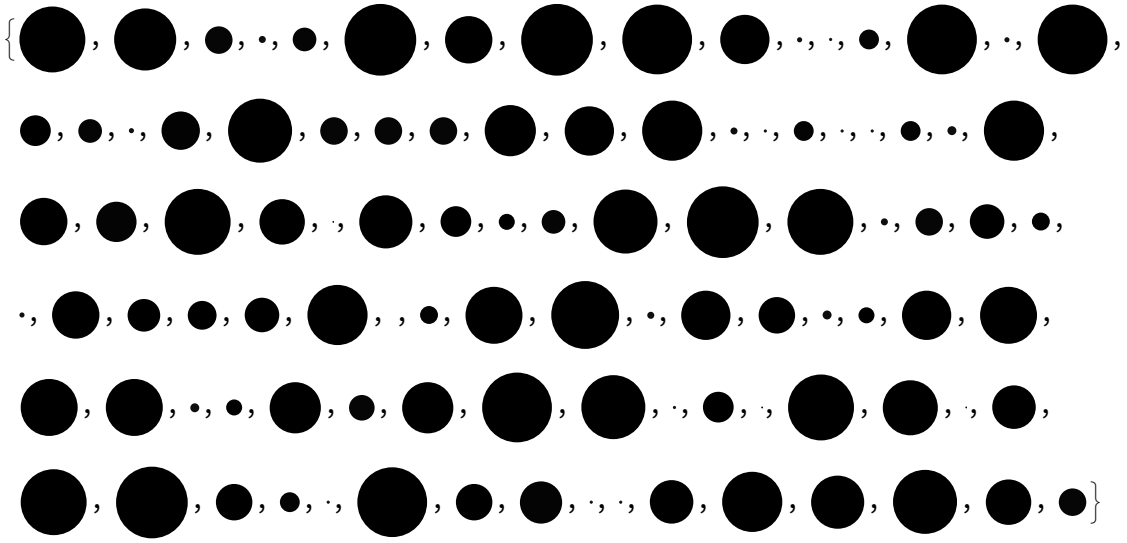
Out[243]=

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[244]:=

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

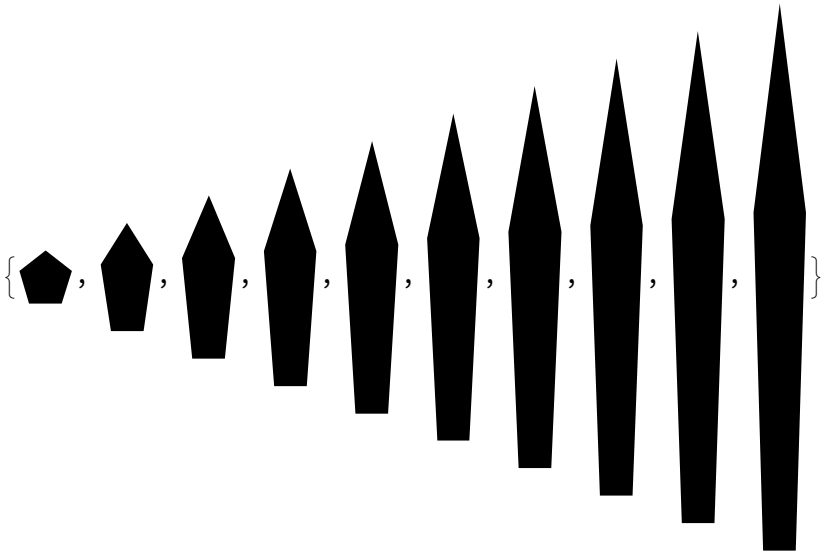
Out[244]=



In[245]:=

```
Table[Graphics[RegularPolygon[5], ImageSize → 30, AspectRatio → x], {x, 1, 10}]
```

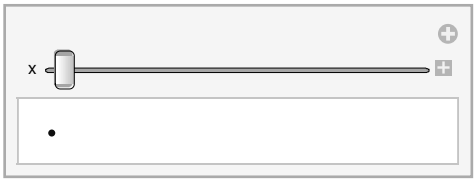
Out[245]=



In[246]:=

```
Manipulate[Graphics[Disk[], ImageSize → x], {x, 5, 500}]
```

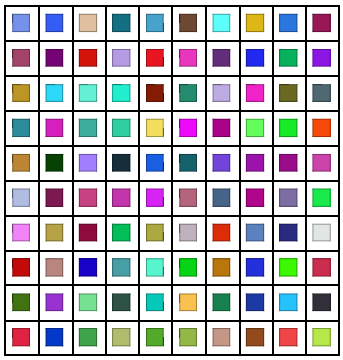
Out[246]=



In[247]:=

```
Grid[Table[RandomColor[], 10, 10], Frame → All]
```

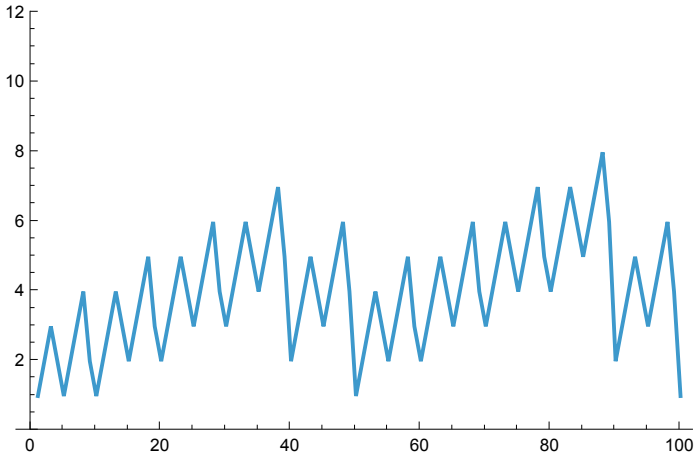
Out[247]=



In[248]:=

```
ListLinePlot[StringLength[RomanNumeral[Range[100]]],
 PlotRange → Max[StringLength[RomanNumeral[Range[1000]]]]]
```

Out[248]=

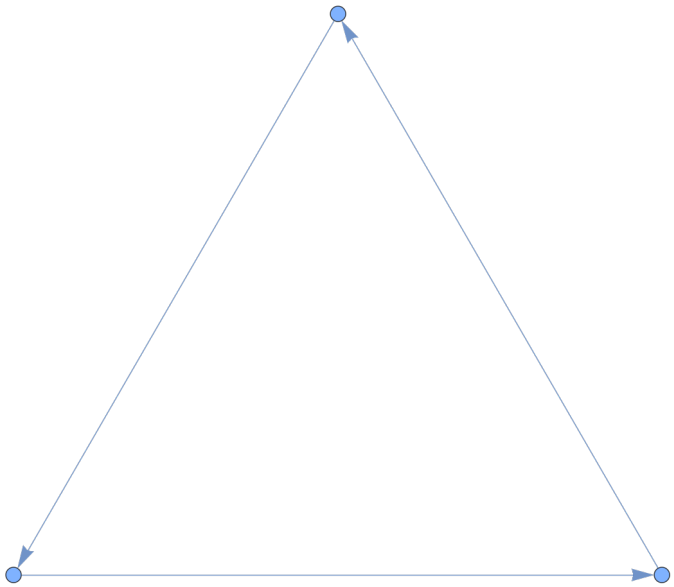


Section 21

In[249]:=

```
Graph[{1 → 2, 2 → 3, 3 → 1}]
```

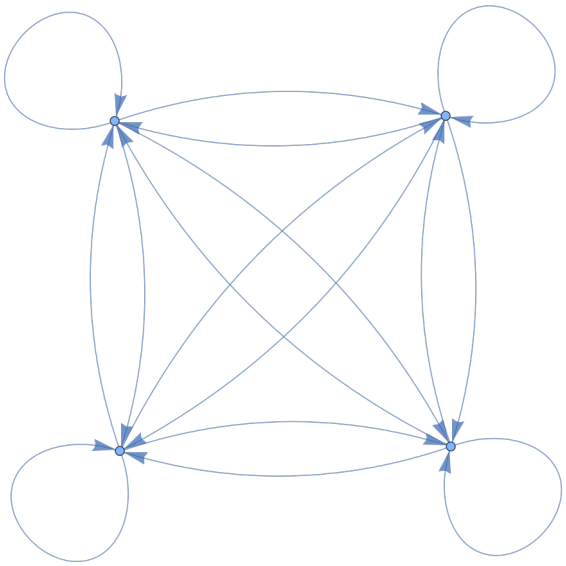
Out[249]=



In[250]:=

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

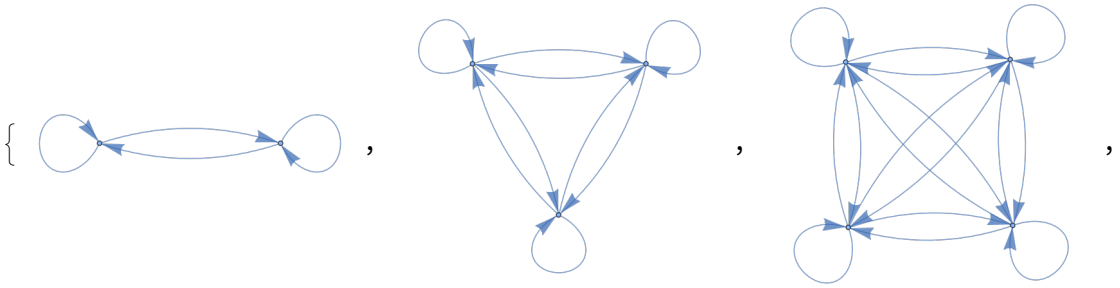
Out[250]=



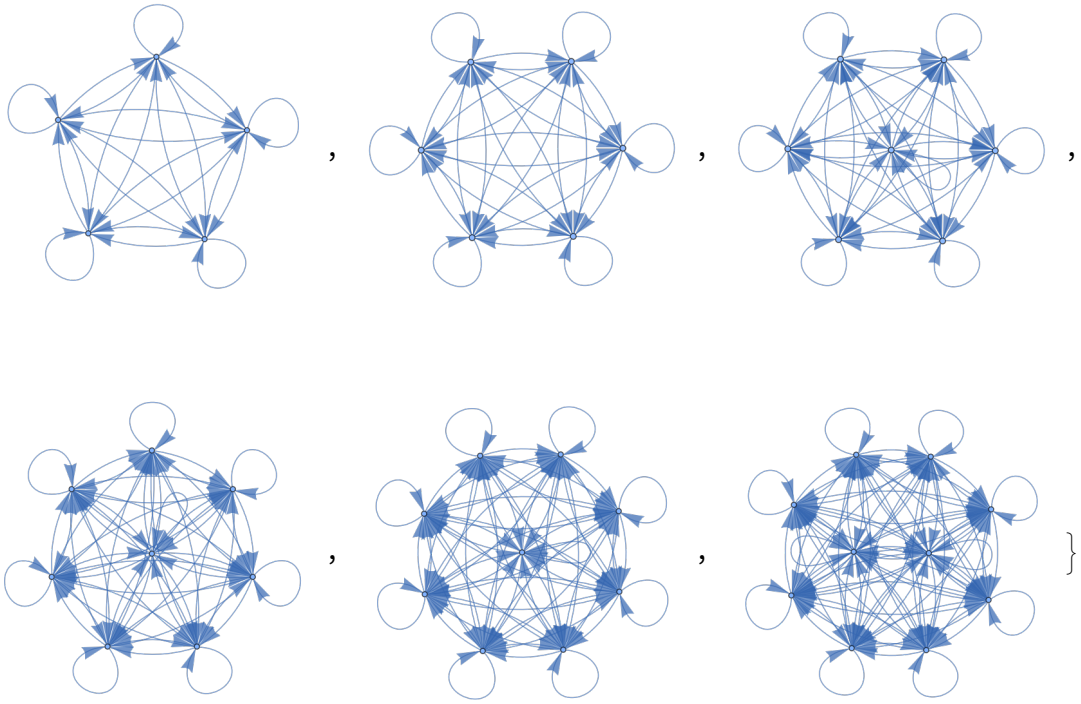
I had a simpler interpretation of what graph he was asking for on this one.

In[251]:=
```
Table[Graph[Flatten[Table[{i → j}, {i, x}, {j, x}]]], {x, 2, 10}]
```

Out[251]=



On this one, he was looking for undirected graphs (without the arrows).

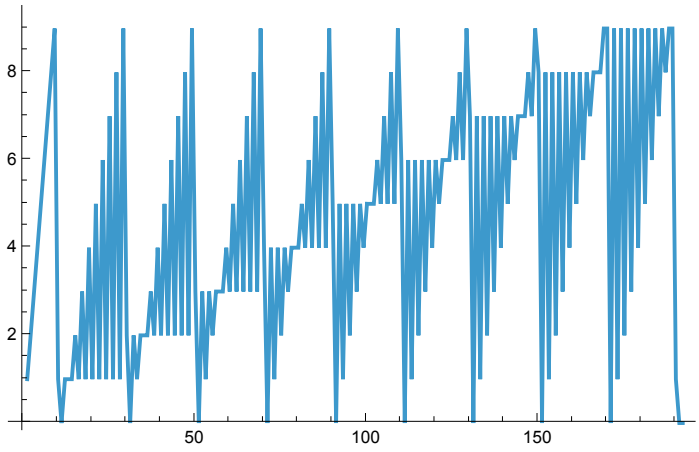


In[252]:=
```
Flatten[Table[{1, 2}, 3]]
```

Out[252]=
```
{1, 2, 1, 2, 1, 2}
```

In[253]:=
```
ListLinePlot[Flatten[IntegerDigits[Range[100]]]]
```

Out[253]=



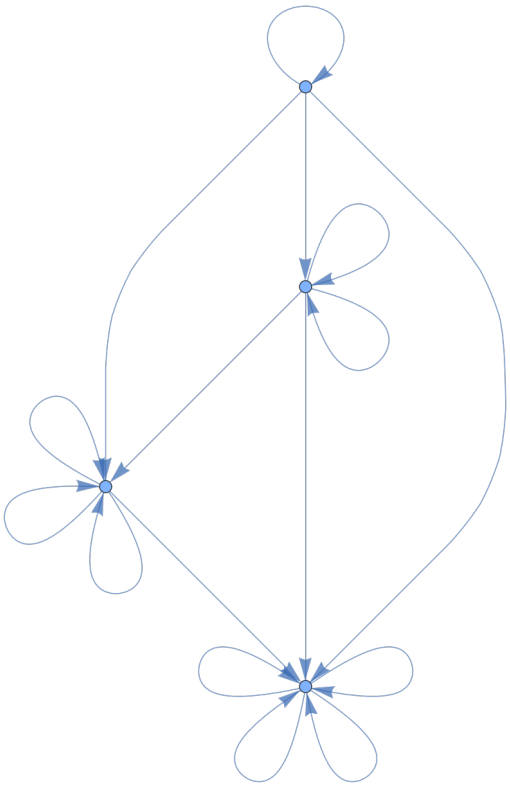
In[254]:=
```
Graph[Flatten[Table[{i → i + 1}, 50, {i, 50}]]]
```

Out[254]=

Oops, not what he was asking for. There is an extra factor of 50 enumerations in your table, which you had to compensate for with by flattening.

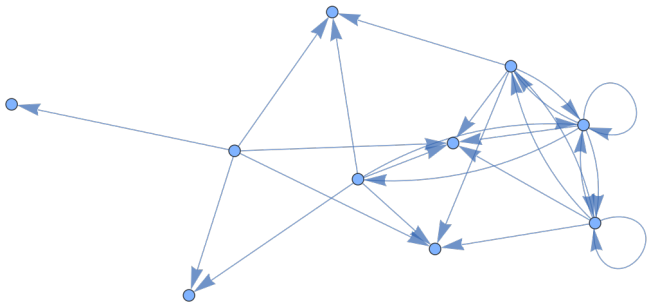
In[255]:=
```
Graph[Flatten[Table[{i → Max[i, j]}, {i, 4}, {j, 4}]]]
```

Out[255]=



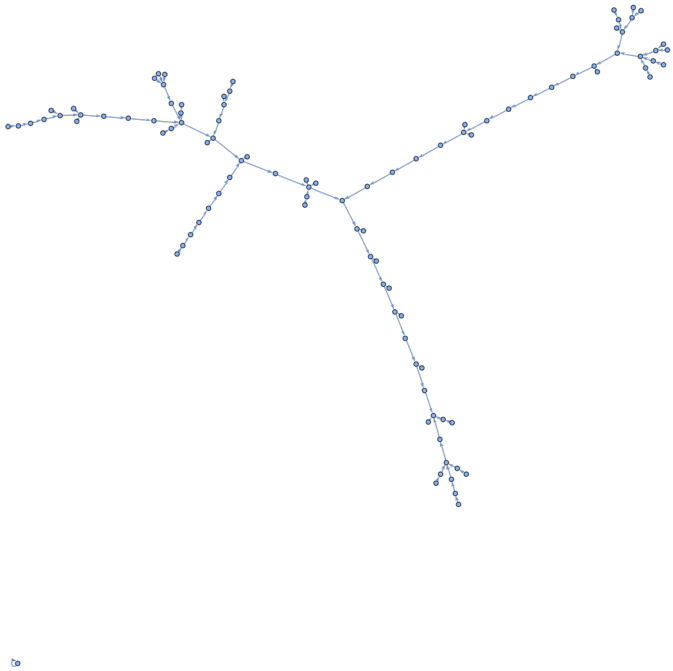
In[256]:=
```
Graph[Flatten[Table[{i → j - i}, {i, 5}, {j, 5}]]]
```

Out[256]=



In[257]:=
```
Graph[Flatten[Table[{i → RandomInteger[100]}, {i, 100}]]]
```

Out[257]=



In[258]:=
```
Grid[Table[FindShortestPath[
   Graph[{1 → 2, 2 → 3, 3 → 4, 4 → 1, 3 → 2, 2 → 2}], x, y], {x, 4}, {y, 4}]]
```

Out[258]=

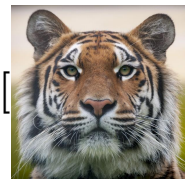
{1}	{1, 2}	{1, 2, 3}	{1, 2, 3, 4}
{2, 3, 4, 1}	{2}	{2, 3}	{2, 3, 4}
{3, 4, 1}	{3, 2}	{3}	{3, 4}
{4, 1}	{4, 1, 2}	{4, 1, 2, 3}	{4}

Section 22

In[259]:=
```
LanguageIdentify["ajatella"]
```
Out[259]=
Finnish

In[260]:=
```
ImageIdentify[ ]
```


```
(*idk what's up with this but i believe the code to be correct*)
```
Out[260]=
tiger

In[261]:=
```
Table[ImageIdentify[Blur[ , r]], {r, 1, 5}]
```


Out[261]=
{tiger, tiger, tiger, tiger, tiger}

For me the 5th blurring it thought was a “swift fox.” My best guess is that you got a different answer only because you started with a different tiger image.

In[262]:=
```
Classify["Sentiment", "I'm so happy to be here"]
```
Out[262]=
Positive

In[263]:=
```
Nearest[WordList[], "happy", 10]
```
Out[263]=
{happy, haply, harpy, nappy, sappy, apply, campy, choppy, guppy, hairy}

In[264]:=

In[265]:=
```
Nearest[Table[RandomInteger[1000], 20], 1000, 3]
```
Out[265]=
{992, 971, 633}

He was looking for the nearest 3 to 100, not 1000.

```
Nearest[Table[RandomColor[], 10], Red ... ✓, 5]
```
Out[266]=
{ , , , , }

In[267]:=

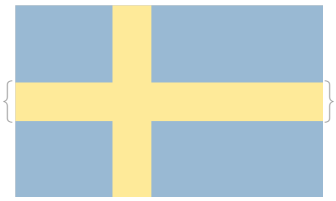
```
Nearest[Table[x^2, {x, 200}], 2000]
```

Out[267]=

```
{2025}
```

```
Nearest[ Europe COUNTRIES [ flag ] ..., Brazil COUNTRY [ flag ] ]
```

Out[268]=

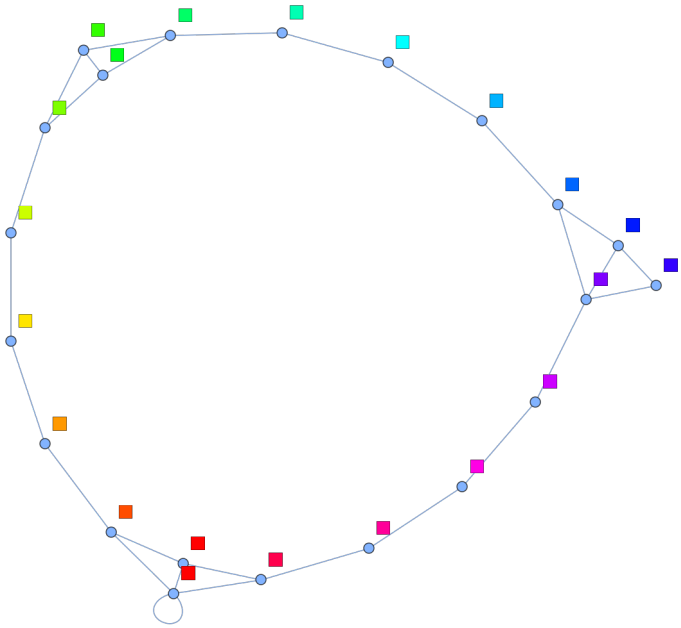


He was looking for the nearest 3.

In[269]:=

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

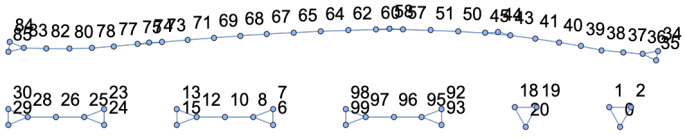
Out[269]=



In[270]:=

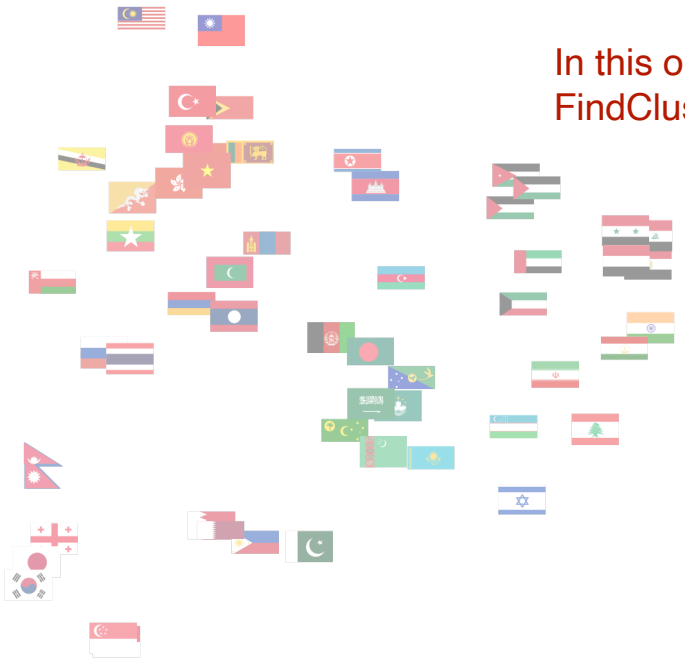
```
NearestNeighborGraph[Table[RandomInteger[100], 100], 2, VertexLabels → All]
```

Out[270]=



```
FeatureSpacePlot[ Asia COUNTRIES [ flag ] ✓ ]
```

Out[271]=

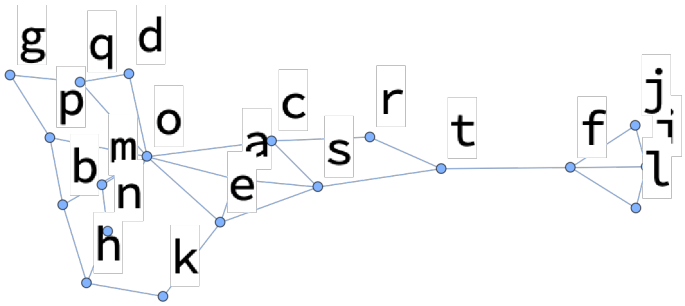


In this one, he was looking for FindClusters, but I like your result too.

In[272]:=

```
NearestNeighborGraph[
 Table[Rasterize[Style[FromLetterNumber[x], 20]], {x, 20}], 2, VertexLabels → All]
```

Out[272]=



In[273]:=

```
Table[TextRecognize[EdgeDetect[Rasterize[Style["programming", x]]]], {x, 10, 20}]
```

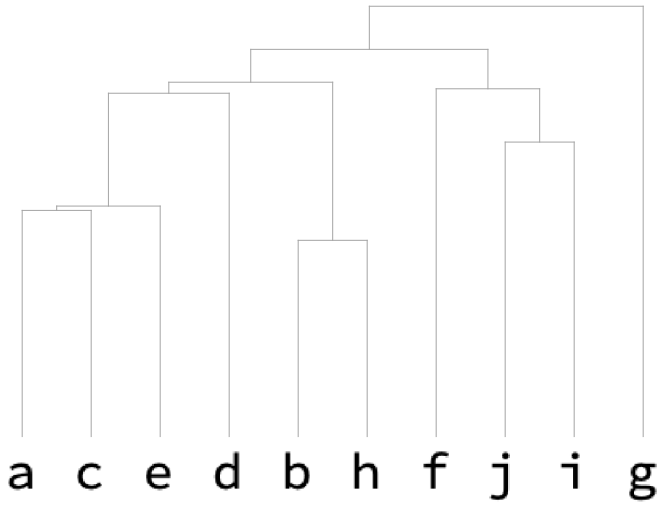
Out[273]=

```
{orogramming, programming, programming, programming, programming,
 programming, programming, programming, programming, programming, programming}
```

In[274]:=

```
Dendrogram[Table[Rasterize[FromLetterNumber[x]], {x, 10}]]
```

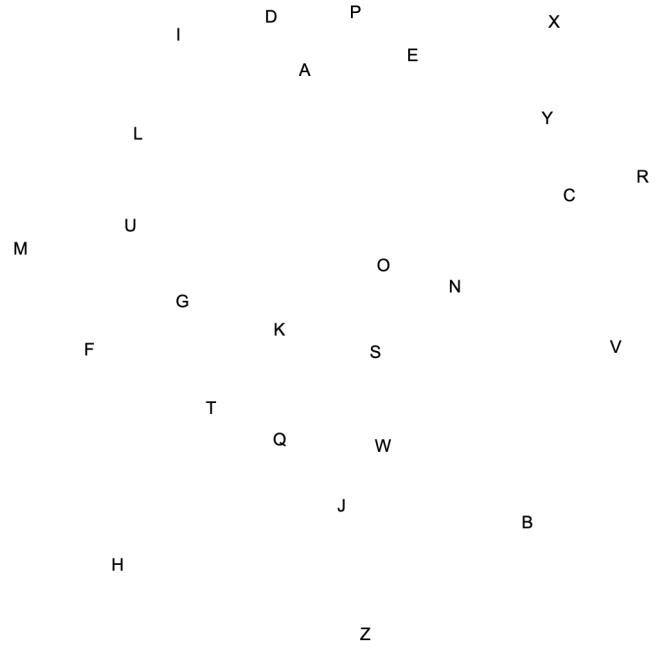
Out[274]=



In[275]:=

```
FeatureSpacePlot[Table[ToUpperCase[FromLetterNumber[x]], {x, 26}]]
```

Out[275]=



This last one isn't right. See my solution. Probably just missing the Rasterize I assume he wanted.