Social Network Analysis

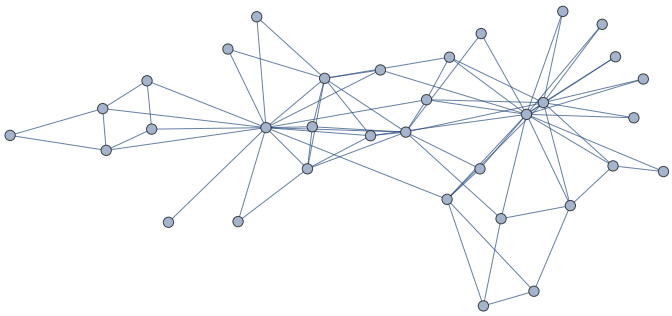
Chen Avin
Communication Systems Engineering

Unit 2
(more) Graph Background & Datasets

Recall from last time

In[1]:=
```
Karate = Graph[ExampleData[{"NetworkGraph", "ZacharyKarateClub"}],
  VertexLabels → Placed["Name", Tooltip]]
```

Out[1]=



Graph Representation

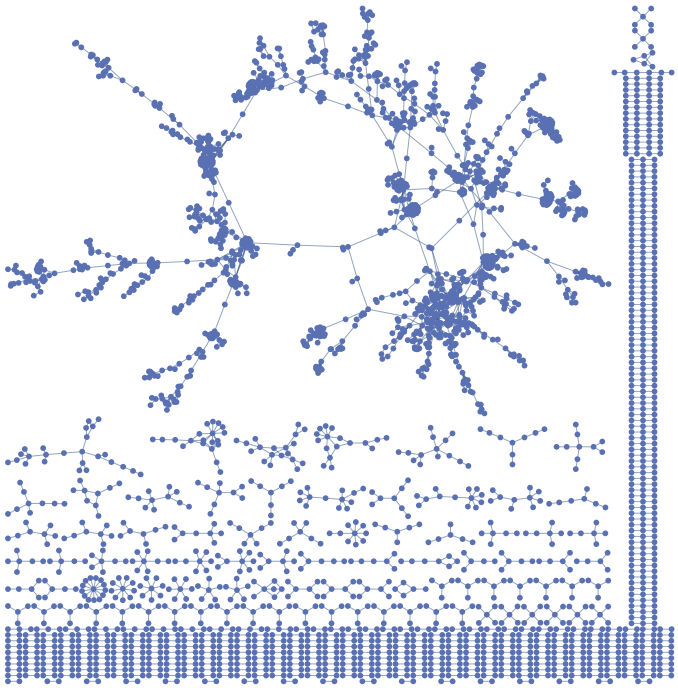
Vertex Degree

Connected Components

- Strongly connected components
- Weakly connected components
- Largest connected components

In[2]:=
```
DN = ExampleData[{"NetworkGraph", "BipartiteDiseasomeNetwork"}]
```

Out[2]=



```
Length[ConnectedComponents[DN]]
```

```
540
```

```
Length[First[ConnectedComponents[DN]]]
```

```
1419
```

```
Length[ConnectedComponents[DN][[2]]]
```

```
25
```

```
VertexCount[DN]
```

```
3061
```

Special Graphs

Complete Graphs

Trees

Grids

Bipartite Graphs

Affiliation Network Example

In[3]:=
```
Marvel = ExampleData[{"NetworkGraph", "MarvelUniverseSocialGraph"}];
```

```
BipartiteGraphQ[Marvel]
```

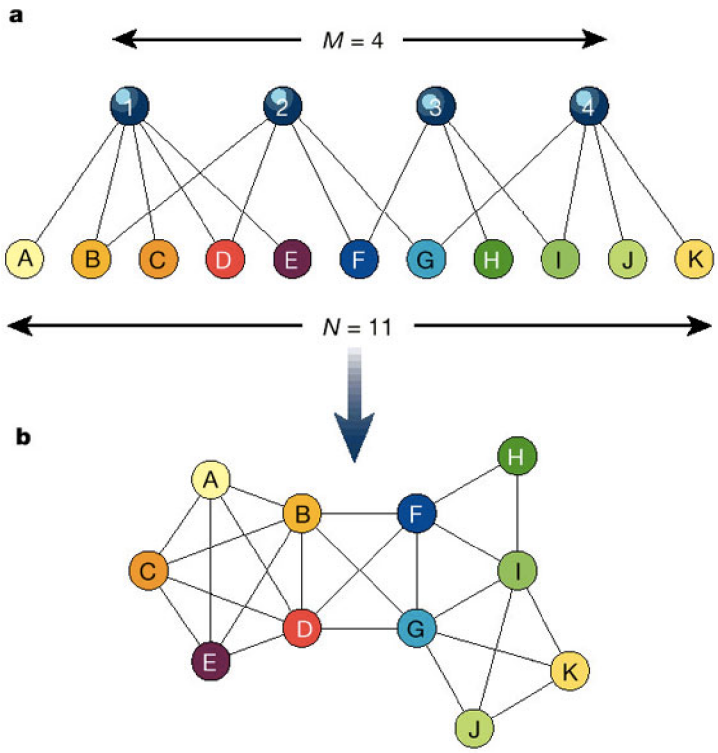
True

```
ExampleData[{"NetworkGraph", "MarvelUniverseSocialGraph"}, "VertexProperty"]
```

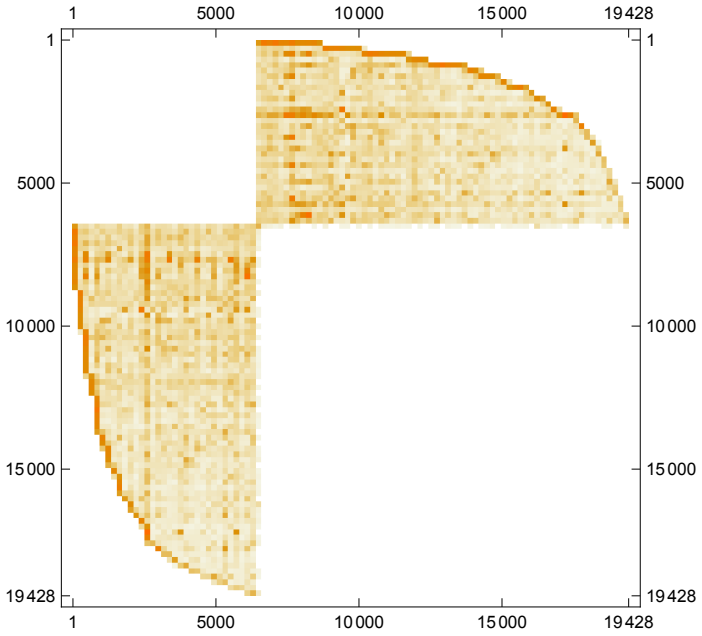
{Name, Type}

```
DeleteDuplicates[PropertyValue[{Marvel, #}, "Type"] & /@ VertexList[Marvel]]
```

{Hero, Comic}




```
MatrixPlot[AdjacencyMatrix[Marvel]]
```



In[4]:=
```
Heros =
  Select[VertexList[Marvel], PropertyValue[{Marvel, #}, "Type"] == "Hero" &];
```

In[ ]:=
```
Length[Heros]
```

Out[ ]=

```
6486
```

In[ ]:=
```
PropertyValue[{Marvel, 6487}, "Type"]
```

Out[ ]=

```
Comic
```

In[5]:=
```
Comics =
  Select[VertexList[Marvel], PropertyValue[{Marvel, #}, "Type"] == "Comic" &];
```

In[•]:=
```
PropertyValue[{Marvel, #}, "Name"] & /@
 Take[Part[VertexList[Marvel], Ordering[VertexDegree[Marvel], All, Greater]], 3]
```

Out[•]=
```
{SPIDER-MAN/PETER PAR, CAPTAIN AMERICA, IRON MAN/TONY STARK }
```

```
VertexDegree[Marvel][[
 Take[Part[VertexList[Marvel], Ordering[VertexDegree[Marvel], All, Greater]], 3]]]
```
```
{1625, 1367, 1168}
```

```
PropertyValue[{Marvel, #}, "Name"] & /@ Take[Part[VertexList[Marvel][[Comics]],
   Ordering[VertexDegree[Marvel][[Comics]], All, Greater]], 3]
```
```
{COC 1, IW 3, IW 1}
```

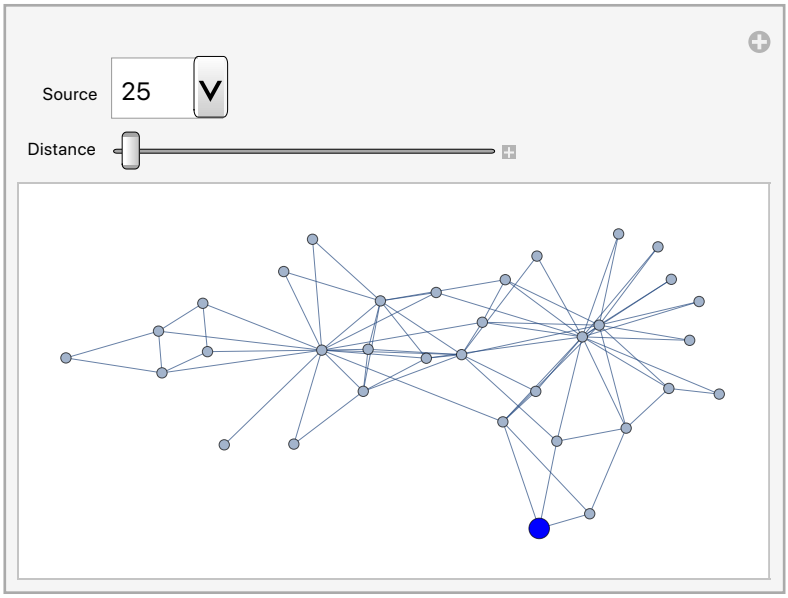
```
VertexDegree[Marvel][[Take[Part[VertexList[Marvel][[Comics]],
   Ordering[VertexDegree[Marvel][[Comics]], All, Greater]], 3]]]
```
```
{111, 91, 90}
```

Distance and Breadth-First Search

Breadth-First Search (BFS) discovers the distances to nodes one layer at a time; each layer is built of nodes that have an edge to at least one node in the previous layer

In[6]:=
```
Manipulate[HighlightGraph[Karate, {Style[s, Blue],
    VertexList[Karate][[Flatten[Position[GraphDistance[Karate, s], d]]]]},
  VertexSize → 1], {{s, 25, "Source"}, Range[34]}, {{d, 0, "Distance"}, 0, 6, 1}]
```

Out[6]=

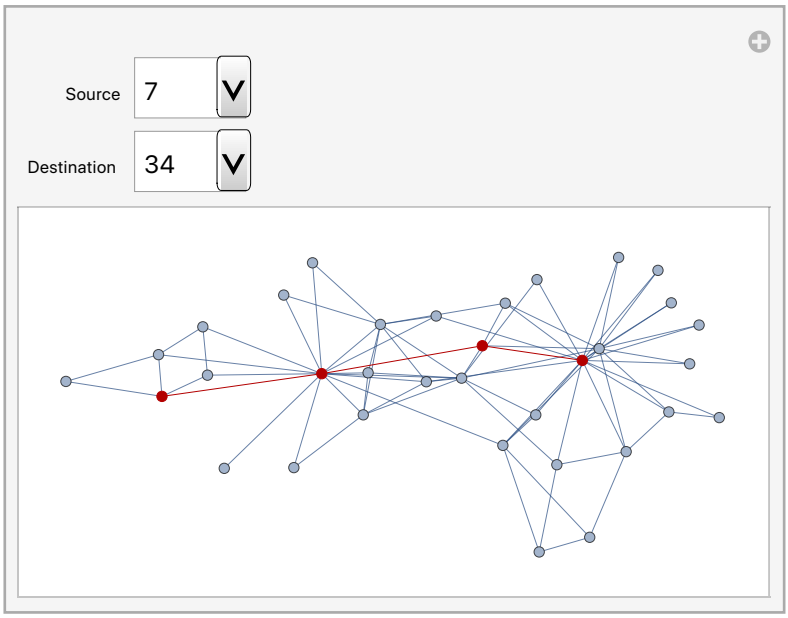


■ Path Graph

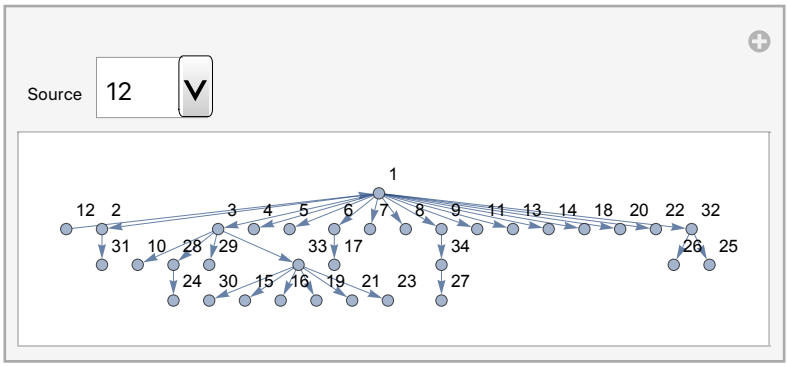
```
PathGraph[FindShortestPath[Karate, 25, 2],
 DirectedEdges → True, VertexLabels → "Name"]
```

■ Shortest Path

```
Manipulate[HighlightGraph[Karate, PathGraph[FindShortestPath[Karate, s, d]]],
 {{s, 1, "Source"}, Range[34]}, {{d, 25, "Destination"}, Range[34]}]
```




```
Manipulate[TreeGraph[
  DeleteDuplicates[Flatten[EdgeList[PathGraph[FindShortestPath[Karate, s, #],
       DirectedEdges → True]] & /@ VertexList[Karate]]],
  VertexLabels → "Name"], {{s, 25, "Source"}, Range[34]}]
```



EdgeList : A graph object is expected at position 1 in
EdgeList [PathGraph [FindShortestPath [Karate, 12, Karate], DirectedEdges → True]].

VertexList : A graph object is expected at position 1 in
VertexList [EdgeList [PathGraph [FindShortestPath [Karate, 12, Karate], DirectedEdges → True]]].

■ Distance Matrix

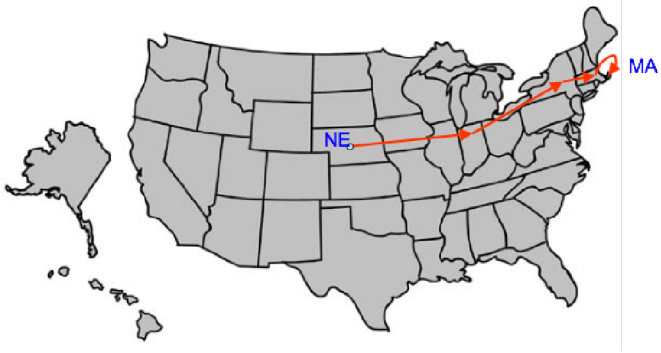
```
In[ ]:= TableForm[GraphDistanceMatrix[Karate],
         TableHeadings → {VertexList[Karate], VertexList[Karate]}]
```

Out[ ]//TableForm=

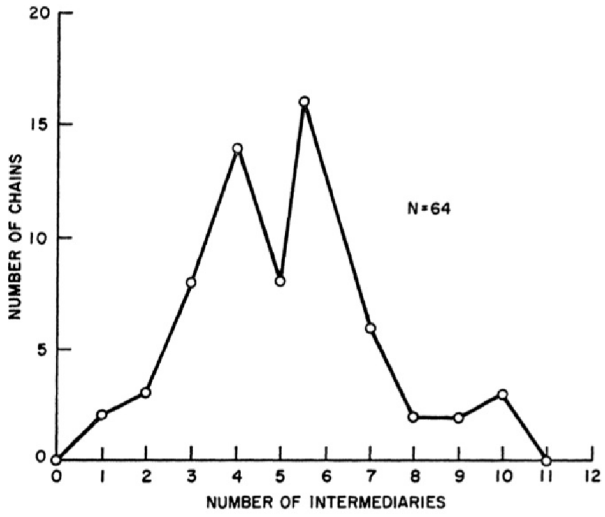
	2	1	3	4	5	6	7	8	9	10	11	12	13	14	17	1
2	0	1	1	1	2	2	2	1	2	2	2	2	2	1	3	1
1	1	0	1	1	1	1	1	1	1	2	1	1	1	1	2	1
3	1	1	0	1	2	2	2	1	1	1	2	2	2	1	3	2
4	1	1	1	0	2	2	2	1	2	2	2	2	1	1	3	2
5	2	1	2	2	0	2	1	2	2	3	1	2	2	2	2	2
6	2	1	2	2	2	0	1	2	2	3	1	2	2	2	1	2
7	2	1	2	2	1	1	0	2	2	3	2	2	2	2	1	2
8	1	1	1	1	2	2	2	0	2	2	2	2	2	2	3	2
9	2	1	1	2	2	2	2	2	0	2	2	2	2	2	3	2
10	2	2	1	2	3	3	3	2	2	0	3	3	3	2	4	3
11	2	1	2	2	1	1	2	2	2	3	0	2	2	2	2	2
12	2	1	2	2	2	2	2	2	2	3	2	0	2	2	3	2
13	2	1	2	1	2	2	2	2	2	3	2	2	0	2	3	2
14	1	1	1	1	2	2	2	2	2	2	2	2	2	0	3	2
17	3	2	3	3	2	1	1	3	3	4	2	3	3	3	0	3
18	1	1	2	2	2	2	2	2	2	3	2	2	2	2	3	0
20	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3	2
22	1	1	2	2	2	2	2	2	2	3	2	2	2	2	3	2
26	3	2	3	3	3	3	3	3	3	3	3	3	3	3	4	3
24	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
25	3	2	2	3	3	3	3	3	3	3	3	3	3	3	4	3
28	2	2	1	2	3	3	3	2	2	2	3	3	3	2	4	3
29	2	2	1	2	3	3	3	2	2	2	3	3	3	2	4	3
30	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
27	3	3	3	3	4	4	4	4	2	2	4	4	4	2	5	4
31	1	2	2	2	3	3	3	2	1	2	3	3	3	2	4	2
32	2	1	2	2	2	2	2	2	2	2	2	2	2	2	3	2
33	2	2	1	2	3	3	3	2	1	2	3	3	3	2	4	3
15	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
16	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
19	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
21	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
23	3	3	2	3	4	4	4	3	2	2	4	4	4	2	5	4
34	2	2	2	2	3	3	3	3	1	1	3	3	3	1	4	3

The Small-World Phenomenon

The small world experiment by Stanly Milgram, 1960



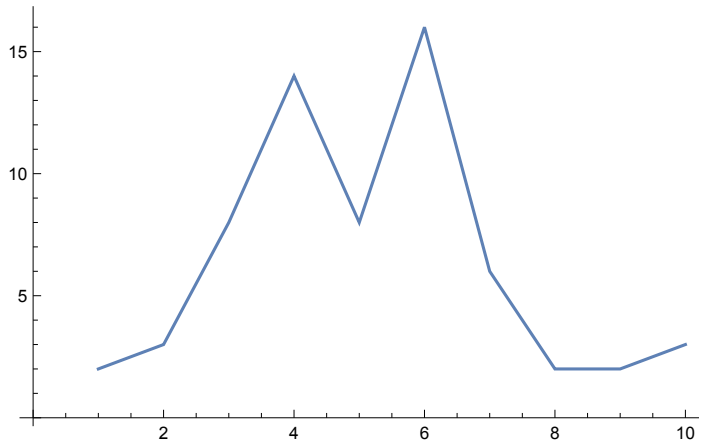
- 296 randomly chosen "starters" to try and forward a letter to target person (stock broker from Boston)
- starters were given some personal information (address & occupation)
- asked to forward the letter on a “first name” basis



In[ ]:=
```
distcahins =
  {{1, 2}, {2, 3}, {3, 8}, {4, 14}, {5, 8}, {6, 16}, {7, 6}, {8, 2}, {9, 2}, {10, 3}};
```

In[ ]:=
```
ListLinePlot[distcahins]
```

Out[ ]=



In[ ]:=
```
chains = Flatten[ConstantArray[#[[1]], #[[2]]] & /@ distcahins]
```

Out[ ]=
```
{1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6,
 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 8, 8, 9, 9, 10, 10, 10}
```

In[ ]:=
```
{Median[chains], Mean[chains]} // N
```

Out[ ]=
```
{5., 5.15625}
```

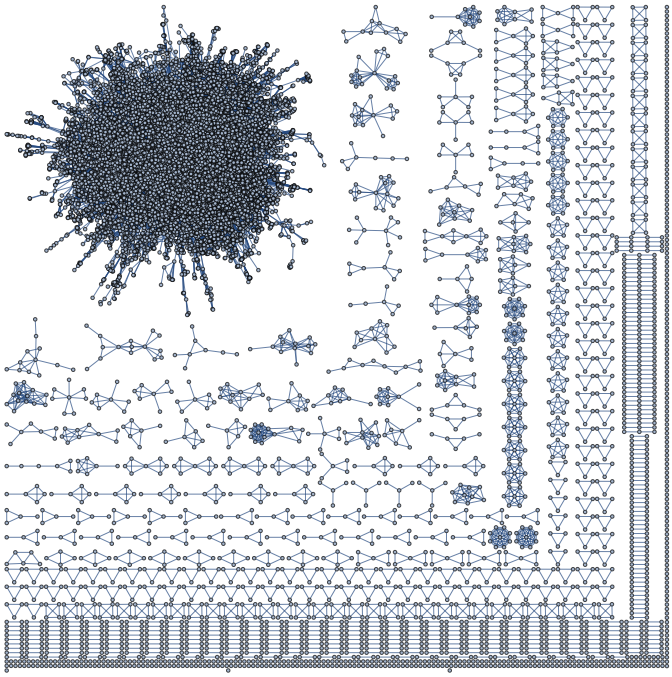
Examples: Small-World

Let's check: network of coauthorships between scientists posting preprints on the Condensed Matter (physics)

In[7]:=
```
SetSystemOptions["GraphOptions" →
    {"EdgeCountThreshold" → 50 000, "VertexCountThreshold" → 20 000}];
```

In[8]:=
```
CM = ExampleData[{"NetworkGraph", "CondensedMatterCollaborations"}]
```

Out[8]=



■ Size

In[ ]:=
```
VertexCount[CM]
```

Out[ ]=

16 726

■ Largest connected component

In[9]:=
```
Length[First[ConnectedComponents[CM]]]
```

Out[9]= 13 861

In[11]:=
```
LccCM = Subgraph[CM, First[ConnectedComponents[CM]]];
```

■ Average Degree

In[12]:=
```
Mean[VertexDegree[LccCM]] // N
```

Out[12]=

6.43806

■ Short Distance Distribution

In[34]:=
```
WeightedGraphQ[LccCM]
```
Out[34]=
```
True
```

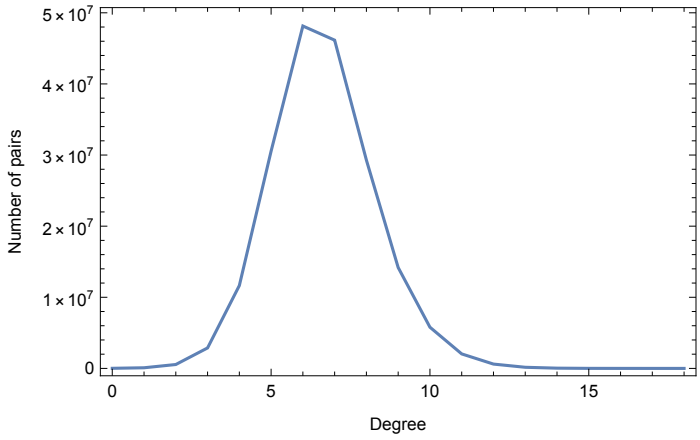
In[35]:=
```
LccCMU = AdjacencyGraph[AdjacencyMatrix[LccCM]];
```

In[37]:=
```
dists = Flatten[GraphDistanceMatrix[LccCMU]];
```

In[38]:=
```
Length[dists]
```
Out[38]=
```
192 127 321
```

In[39]:=
```
TD = Tally[dists]
```
Out[39]=
```
{{0, 13 861}, {1, 89 238}, {2, 548 114}, {3, 2 878 214}, {4, 11 648 306},
 {5, 30 548 968}, {6, 48 141 174}, {7, 46 151 780}, {8, 29 284 594},
 {9, 14 184 534}, {10, 5 792 944}, {11, 2 037 414}, {12, 607 982},
 {13, 158 128}, {14, 34 828}, {15, 6190}, {16, 890}, {17, 132}, {18, 30}}
```

In[40]:=
```
ListPlot[Sort[TD], Joined → True, Frame → True,
 FrameLabel → {"Degree", "Number of pairs"}]
```
Out[40]=



In[41]:=
```
{Median[dists], Mean[dists]} // N
```
Out[41]=
```
{7., 6.62735}
```

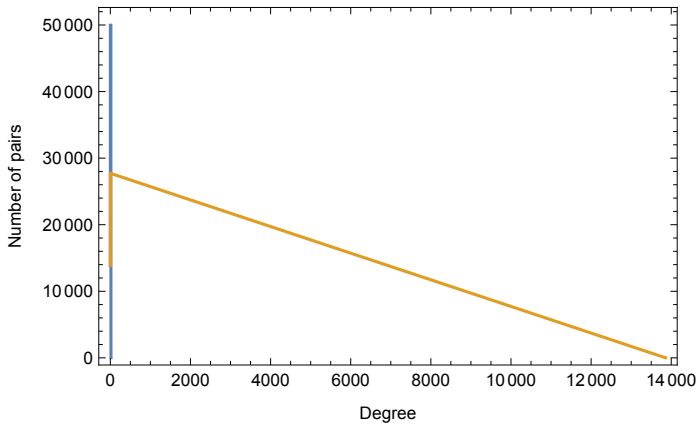
Are all networks small world???

In[42]:=
```
GridGraph[{30}]
```
Out[42]=

In[43]:=
```
distG1 = GraphDistanceMatrix[GridGraph[{13 861}]];
```

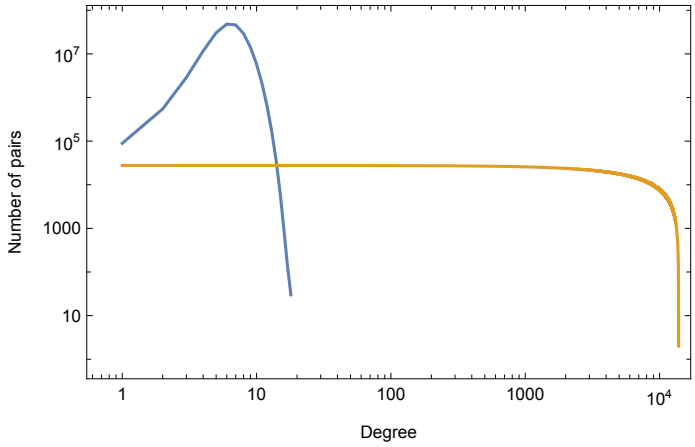
In[66]:=
```
ListPlot[{Sort[TD], Sort[Tally[Flatten[distG1]]]}, Joined → True,
 Frame → True, FrameLabel → {"Degree", "Number of pairs"}]
```

Out[66]=

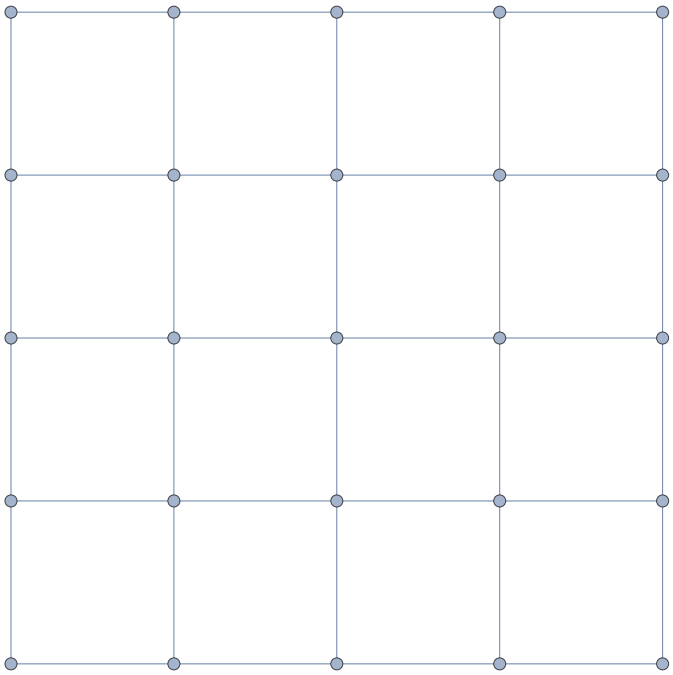


In[67]:=
```
ListLogLogPlot[{Sort[TD], Sort[Tally[Flatten[distG1]]]}, Joined → True,
 PlotRange → All, Frame → True, FrameLabel → {"Degree", "Number of pairs"}]
```

Out[67]=



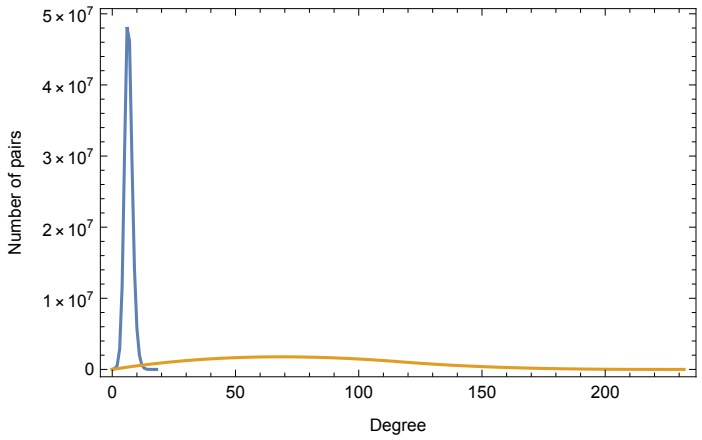
```
GridGraph[{5, 5}]
```



In[68]:=
```
distG2 =
  GraphDistanceMatrix[GridGraph[{Floor[Sqrt[13 861]], Floor[Sqrt[13 861]]}]];
```

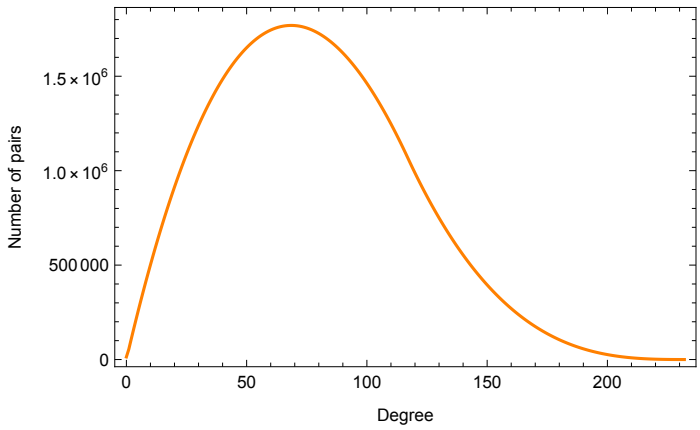
In[69]:=
```
ListPlot[{Sort[Tally[dists]], Sort[Tally[Flatten[distG2]]]}, Joined → True,
 PlotRange → All, Frame → True, FrameLabel → {"Degree", "Number of pairs"}]
```

Out[69]=



In[71]:=
```
ListPlot[Sort[Tally[Flatten[distG2]]], Joined → True, PlotRange → All,
 Frame → True, FrameLabel → {"Degree", "Number of pairs"}, PlotStyle → Orange]
```

Out[71]=



In[72]:=
```
Mean[Flatten[distG1]] // N
```

Out[72]=
```
4620.33
```

In[73]:=
```
Mean[Flatten[distG2]] // N
```

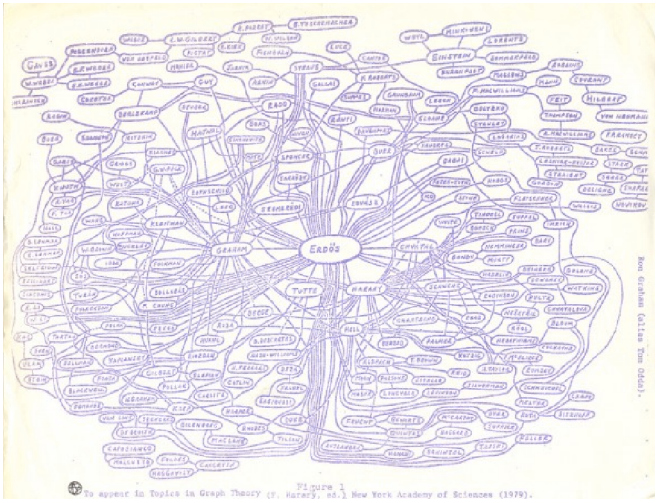
Out[73]=
```
77.9943
```

In[74]:=
```
Floor[Sqrt[13 861]]
```

Out[74]=
```
117
```

Some On - line Examples

- Erdos Number Project - http://wwwp.oakland.edu/enp/



- https://www.csauthors.net/distance/
- The Oracle of Bacon - https://oracleofbacon.org/
- See! It's a small world (in BGU) - http://nap.cse.bgu.ac.il/sa/

Is the world getting larger or Smaller ?

check Facebook

DataSets and Networks

In[75]:=
```
NetList = ExampleData["NetworkGraph"];
```

In[76]:=
```
Length[NetList]
```

Out[76]=
```
228
```

In[77]:=
```
NetList[[All, 2]]
```

Out[77]=
```
{AmericanCollegeFootball, AskOpinionRecall, AskOpinionRecognition,
 AstrophysicsCollaborations, BeAskedOpinionRecall, BeAskedOpinionRecognition,
 BipartiteDiseasomeNetwork, Brock2001, Brock2002, Brock2003, Brock2004,
 Brock4001, Brock4002, Brock4003, Brock4004, Brock8001, Brock8002, Brock8003,
 Brock8004, BuddingYeast, CellOntology, CFat2001, CFat2002, CFat2005,
 CFat5001, CFat50010, CFat5002, CFat5005, CoauthorshipsInNetworkScience,
 CondensedMatterCollaborations, CondensedMatterCollaborations2003,
 CondensedMatterCollaborations2005, DavisSouthernWomen, DiscussionRecall,
 DiscussionRecognition, DiseaseGeneNetwork, DolphinSocialNetwork,
 EastAfricaEmbassyAttacks, EmailListMathGroup, EuclidElements,
 EurovisionVotes, ExpandedAbortion, ExpandedComputationalComplexity,
 ExpandedComputationalGeometry, ExpandedDeathPenalty, ExpandedGenetic,
 ExpandedGunControl, ExpandedMovies, ExpandedNetCensorship,
 FamilyGathering, FlorentineFamilies, FreeAssociationNormsAppendixA,
 Friendship, Hamming102, Hamming104, Hamming62, Hamming64, Hamming82,
 Hamming84, HighEnergyPhysicsPhenomenology, HighEnergyPhysicsTheory,
 HighEnergyTheoryCollaborations, HumanDiseaseNetwork, Internet, JazzMusicians,
 Johnson1624, Johnson3224, Johnson824, Johnson844, Keller4, Keller5, Keller6,
 LesMiserables, MannA27, MannA45, MannA81, MannA9, MarvelUniverseSocialGraph,
 MetabolicNetworkActinobacillusActinomycetemcomitans,
 MetabolicNetworkAeropyrumPernix, MetabolicNetworkAquifexAeolicus,
 MetabolicNetworkArabidopsisThaliana, MetabolicNetworkArchaeoglobusFulgidus,
 MetabolicNetworkBacillusSubtilis, MetabolicNetworkBorreliaBurgdorferi,
 MetabolicNetworkCaenorhabditisElegans, MetabolicNetworkCampylobacterJejuni,
 MetabolicNetworkChlamydiaPneumoniae, MetabolicNetworkChlamydiaTrachomatis,
 MetabolicNetworkChlorobiumTepidum, MetabolicNetworkClostridiumAcetobutylicum,
 MetabolicNetworkDeinococcusRadiodurans, MetabolicNetworkEmericellaNidulans,
 MetabolicNetworkEnterococcusFaecalis, MetabolicNetworkEscherichiaColi,
 MetabolicNetworkHaemophilusInfluenzae, MetabolicNetworkHelicobacterPylori,
 MetabolicNetworkMethanobacteriumThermoautotrophicum,
 MetabolicNetworkMethanococcusJannaschii,
 MetabolicNetworkMycobacteriumBovis, MetabolicNetworkMycobacteriumLeprae,
 MetabolicNetworkMycobacteriumTuberculosis,
 MetabolicNetworkMycoplasmaGenitalium, MetabolicNetworkMycoplasmaPneumoniae,
 MetabolicNetworkNeisseriaGonorrhoeae, MetabolicNetworkNeisseriaMeningitidis,
 MetabolicNetworkOryzaSativa, MetabolicNetworkPorphyromonasGingivalis,
```

MetabolicNetworkPseudomonasAeruginosa, MetabolicNetworkPyrococcusFuriosus,
MetabolicNetworkPyrococcusHorikoshii, MetabolicNetworkRhodobacterCapsulatus,
MetabolicNetworkRickettsiaProwazekii, MetabolicNetworkSaccharomycesCerevisiae,
MetabolicNetworkSalmonellaTyphi, MetabolicNetworkStreptococcusPneumoniae,
MetabolicNetworkStreptococcusPyogenes, MetabolicNetworkSynechocystisSp,
MetabolicNetworkThermotogaMaritima, MetabolicNetworkTreponemaPallidum,
MetabolicNetworkYersiniaPestis, NationalHockeyLeague, OnlineSocialNetwork,
PerlModuleAuthors, PGPNetwork, PHat10001, PHat10002, PHat10003, PHat15001,
PHat15002, PHat15003, PHat3001, PHat3002, PHat3003, PHat5001, PHat5002,
PHat5003, PHat7001, PHat7002, PHat7003, PoliticalBlogs, PowerGrid,
ProteinInteraction, RefinedAbortion, RefinedComputationalComplexity,
RefinedComputationalGeometry, RefinedDeathPenalty, RefinedGenetic,
RefinedGunControl, RefinedMovies, RefinedNetCensorship, RegularAbortion,
RegularComputationalComplexity, RegularComputationalGeometry,
RegularDeathPenalty, RegularGenetic, RegularGunControl, RegularMovies,
RegularNetCensorship, San1000, San200071, San200072, San200091, San200092,
San200093, San400051, San400071, San400072, San400073, San400091, Sanr20007,
Sanr20009, Sanr40005, Sanr40007, September11Terrorists, SimpleFoodWeb,
SloveneParliamentaryParties, TaggedTestImages, URVEmailNetwork,
USPoliticsBooks, WholeNetworkActinobacillusActinomycetemcomitans,
WholeNetworkAeropyrumPernix, WholeNetworkAquifexAeolicus,
WholeNetworkArabidopsisThaliana, WholeNetworkArchaeoglobusFulgidus,
WholeNetworkBacillusSubtilis, WholeNetworkBorreliaBurgdorferi,
WholeNetworkCaenorhabditisElegans, WholeNetworkCampylobacterJejuni,
WholeNetworkChlamydiaPneumoniae, WholeNetworkChlamydiaTrachomatis,
WholeNetworkChlorobiumTepidum, WholeNetworkClostridiumAcetobutylicum,
WholeNetworkDeinococcusRadiodurans, WholeNetworkEmericellaNidulans,
WholeNetworkEnterococcusFaecalis, WholeNetworkEscherichiaColi,
WholeNetworkHaemophilusInfluenzae, WholeNetworkHelicobacterPylori,
WholeNetworkMethanobacteriumThermoautotrophicum,
WholeNetworkMethanococcusJannaschii, WholeNetworkMycobacteriumBovis,
WholeNetworkMycobacteriumLeprae, WholeNetworkMycobacteriumTuberculosis,
WholeNetworkMycoplasmaGenitalium, WholeNetworkMycoplasmaPneumoniae,
WholeNetworkNeisseriaGonorrhoeae, WholeNetworkNeisseriaMeningitidis,
WholeNetworkOryzaSativa, WholeNetworkPorphyromonasGingivalis,
WholeNetworkPseudomonasAeruginosa, WholeNetworkPyrococcusFuriosus,
WholeNetworkPyrococcusHorikoshii, WholeNetworkRhodobacterCapsulatus,
WholeNetworkRickettsiaProwazekii, WholeNetworkSaccharomycesCerevisiae,
WholeNetworkSalmonellaTyphi, WholeNetworkStreptococcusPneumoniae,
WholeNetworkStreptococcusPyogenes, WholeNetworkSynechocystisSp,
WholeNetworkThermotogaMaritima, WholeNetworkTreponemaPallidum,
WholeNetworkYersiniaPestis, WikiVote, WordAdjacencies,
WorldCup1988, WorldWideWeb, ZacharyKarateClub}

Collaboration Graphs

In[78]:=
```
ExampleData[
 {"NetworkGraph", "CoauthorshipsInNetworkScience"}, "LongDescription"]
```

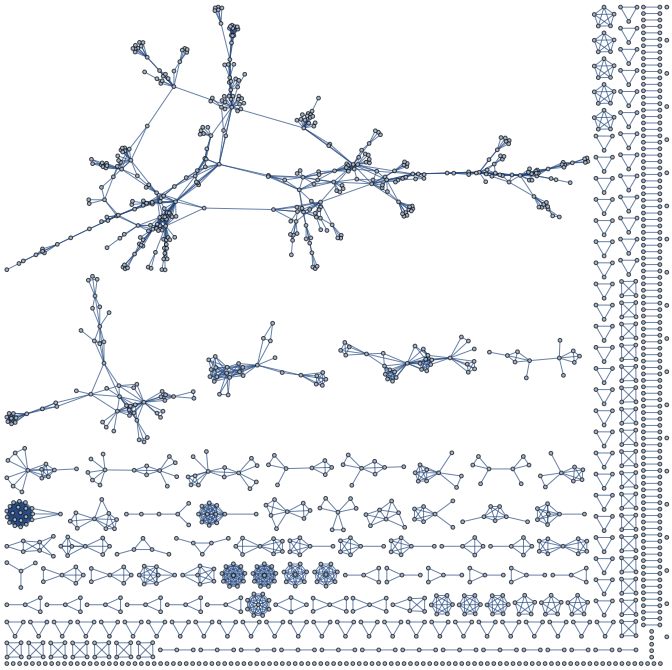
Out[78]=
```
Coauthorship network of scientists working on network theory and experiment,
  as compiled by M. Newman in May 2006. The edge weight indicates
  the strength of the collaboration between pairs of individuals.
```

In[79]:=
```
ExampleData[{"NetworkGraph", "CoauthorshipsInNetworkScience"}, "VertexCount"]
```

Out[79]=
```
1589
```

In[80]:=
```
G = ExampleData[{"NetworkGraph", "CoauthorshipsInNetworkScience"}]
```

Out[80]=



In[81]:=
```
WeightedGraphQ[G]
```

Out[81]=
```
True
```

In[82]:=
```
ExampleData[{"NetworkGraph", "WikiVote"}, "LongDescription"]
```

Out[82]=
```
Network of Wikipedia admins and users who have been
  active in the voting process for new admins.  A directed
  edge from s to t means that s voted for t to become an admin.
```

In[83]:=
```
ExampleData[{"NetworkGraph", "WikiVote"}, "VertexCount"]
```

Out[83]=
```
7115
```

Who-Talks-to-Whom Graphs

In[84]:=
```
ExampleData[{"NetworkGraph", "URVEmailNetwork"}, "LongDescription"]
```

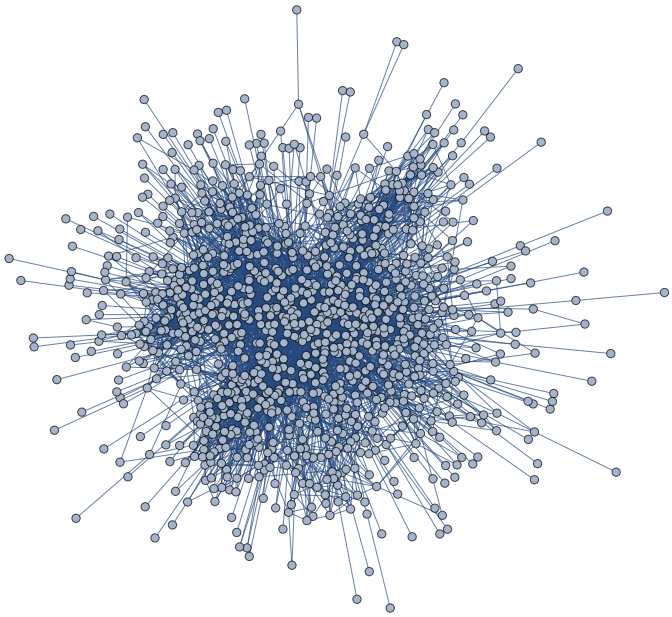
Out[84]=
```
The network of e-mail interchanges between
  members of the Univeristy Rovira i Virgili (Tarragona).
```

In[85]:=
```
ExampleData[{"NetworkGraph", "URVEmailNetwork"}, "VertexCount"]
```

Out[85]=
```
1133
```

In[86]:=
```
ExampleData[{"NetworkGraph", "URVEmailNetwork"}]
```

Out[86]=



Information Linkage Graphs (Web Graph)

In[87]:=
```
ExampleData[{"NetworkGraph", "WorldWideWeb"}, "LongDescription"]
```

Out[87]=

The World-Wide-Web graph of web pages within nd.edu
domain. The edge weight indicates the multiplicity of links.

In[88]:=
```
ExampleData[{"NetworkGraph", "WorldWideWeb"}, "VertexCount"]
```

Out[88]=

325 729

Technological Networks

In[90]:=
```
ExampleData[{"NetworkGraph", "Internet"}, "LongDescription"]
```
Out[90]=
```
A symmetrized snapshot of the structure of the Internet
  at the level of autonomous systems, reconstructed from BGP
  tables posted by the University of Oregon Route Views Project.
```

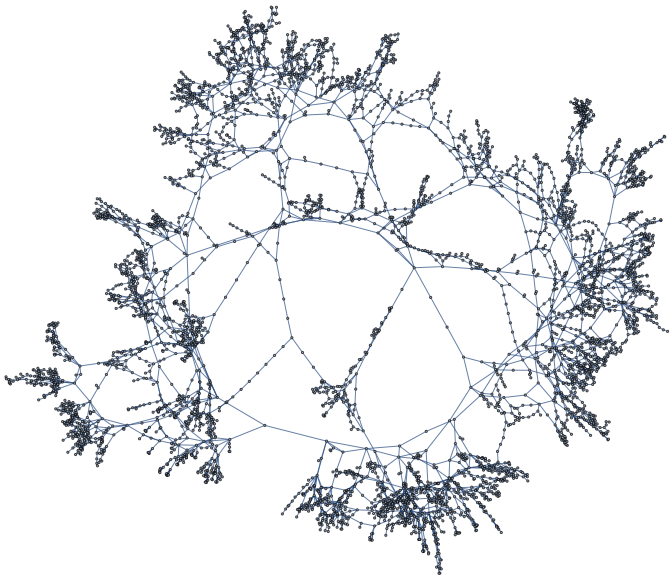
In[91]:=
```
ExampleData[{"NetworkGraph", "Internet"}, "VertexCount"]
```
Out[91]=
```
22 963
```

In[92]:=
```
ExampleData[{"NetworkGraph", "PowerGrid"}, "LongDescription"]
```
Out[92]=
```
An undirected, unweighted network representing the
  topology of the Western States Power Grid of the United States.
```

In[93]:=
```
ExampleData[{"NetworkGraph", "PowerGrid"}, "VertexCount"]
```
Out[93]=
```
4941
```

In[94]:=
```
ExampleData[{"NetworkGraph", "PowerGrid"}, "VertexProperty"]
```
Out[94]=
```
{}
```

In[95]:=
```
ExampleData[{"NetworkGraph", "PowerGrid"}]
```
Out[95]=



Networks in the Natural World

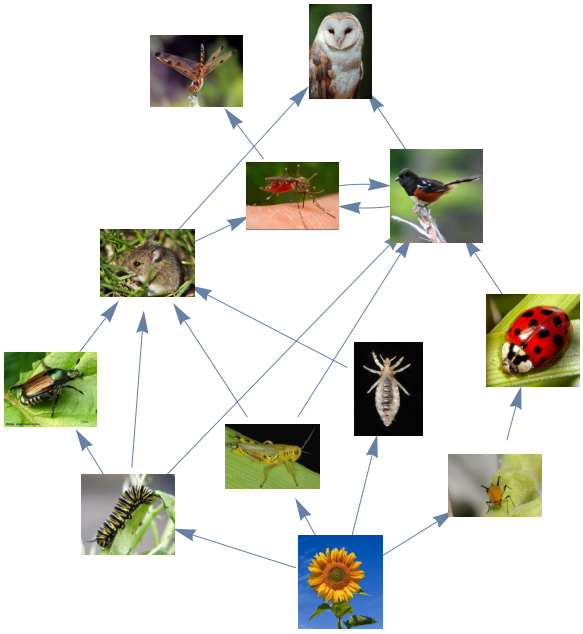
In[96]:=
```
ExampleData[{"NetworkGraph", "SimpleFoodWeb"}, "LongDescription"]
```

Out[96]=

A simple food web. An edge from s to
  t means that the species s is a food source for t.

In[97]:=
```
ExampleData[{"NetworkGraph", "SimpleFoodWeb"}]
```

Out[97]=



In[98]:=
```
ExampleData[
 {"NetworkGraph", "WholeNetworkCaenorhabditisElegans"}, "LongDescription"]
```

Out[98]=

Whole cellular network data for Caenorhabditis Elegans

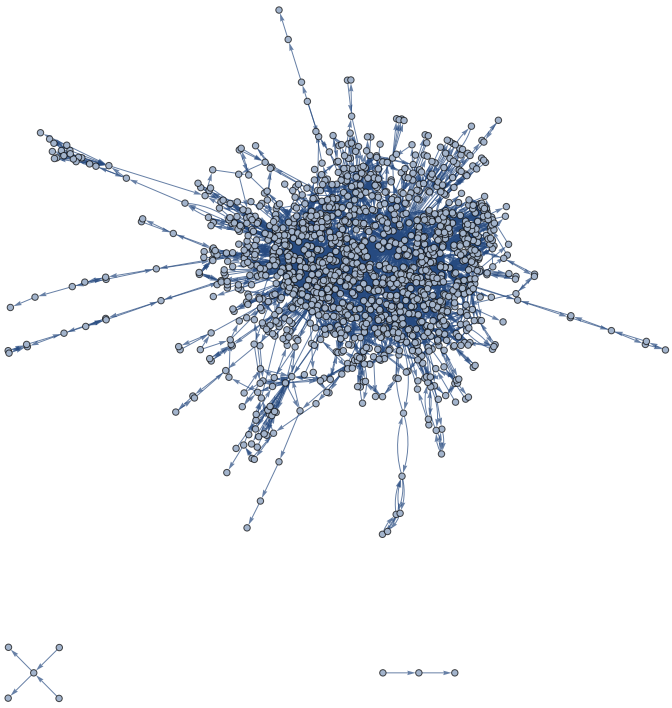
https : // en.wikipedia.org/wiki/Caenorhabditis_elegans

In[99]:=
```
ExampleData[
 {"NetworkGraph", "WholeNetworkCaenorhabditisElegans"}, "VertexCount"]
```

Out[99]=

1469

In[100]:=
```
ExampleData[{"NetworkGraph", "WholeNetworkCaenorhabditisElegans"}]
```
Out[100]=



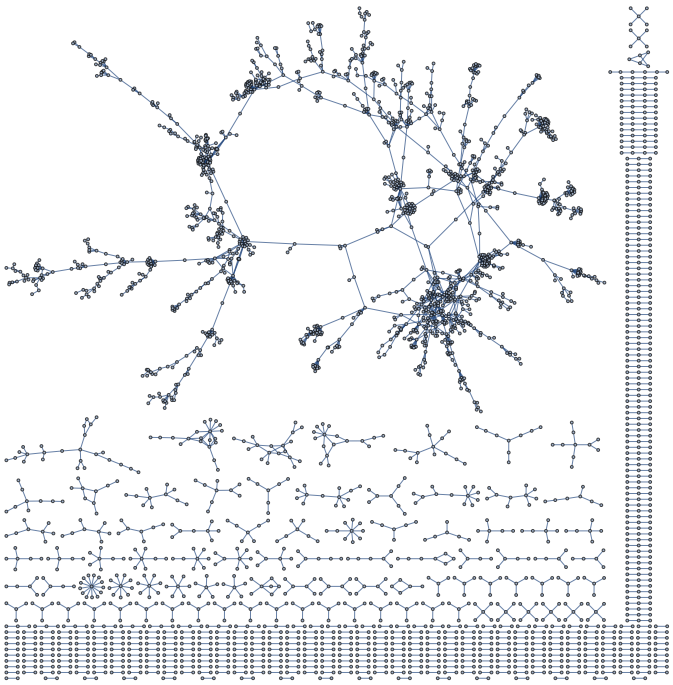
In[101]:=
```
ExampleData[{"NetworkGraph", "BipartiteDiseasomeNetwork"}, "LongDescription"]
```
Out[101]=
```
A network of disorders and disease genes linked by known
  disorderâgene associations, indicating the common genetic origin
  of many diseases. Genes associated with similar disorders show
  both higher likelihood of physical interactions between their
  products and higher expression profiling similarity for their
  transcripts, supporting the existence of distinct disease-specific
  functional modules. Vertex properties contains "Type" and "ID".
```

In[102]:=
```
ExampleData[{"NetworkGraph", "BipartiteDiseasomeNetwork"}, "VertexProperty"]
```
Out[102]=
```
{Type, ID}
```

In[103]:=

```
ExampleData[{"NetworkGraph", "BipartiteDiseasomeNetwork"}]
```

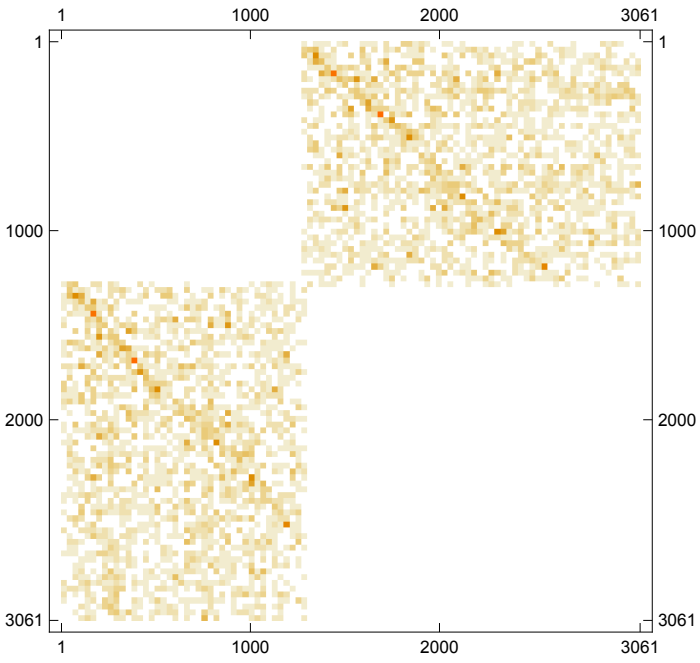
Out[103]=



In[104]:=

```
MatrixPlot[
 AdjacencyMatrix[ExampleData[{"NetworkGraph", "BipartiteDiseasomeNetwork"}]]]
```

Out[104]=



Some Other Sources

- Stanford Large Network Dataset Collection
- The Koblenz Network Collection

Class Assignment Week 2 - 21/3/2023

```
ExampleData[{"NetworkGraph", "MarvelUniverseSocialGraph"}, "LongDescription"]
```

```
The Marvel Universe is an artificial social network that
  pretends to imitate a real social graph, but is closer to a real
  social graph than one might expect. A vertex property "Name"
  describes either the name of hero or the the title of comic
  book.  A vertex property "Type" describes the type of vertex.
```

- Marvel “Affiliation Network”
- Create (nicely) a Hero-to-Hero weighted network - Think how to
  - An edge weight is how many time two heroes appear in the same comic book (story)
- For an input x, show the largest connected component of all node with edge weight larger than x.
- Consider for simplicity x = 10,20,30,40
- For each such network highlight the top nodes with highest degree (think what is top, there could be several options).
- Submit your notebook by Saturday
  - I’m here for questions
- You can work in groups of at most 2 students