[**ICS 311**](http://www2.hawaii.edu/~suthers/courses/ics311s14/index.html)**Project 3: Network Metrics**

**Due by 23:55 (11:55 pm) Saturday May 10th. 100 points.**

Late submissions with **penalty of 1% per hour** accepted until 2:00 **AM** May 13th (50 hours = 50% off). But it is a good idea to finish Saturday so you have Mother's Day off!

Requirements stated on the [Assignments Page](http://www2.hawaii.edu/~suthers/courses/ics311s14/Syllabus/Assignments.html) are included in the requirements for this project.

**Change Log**

May 2, 2014

Reference manual sections on your algorithms should tell us where to find the code for each algorithm so we don't waste time searching for it. Also put pseudocode or a copy of the code directly in the reference manual to refer to in your analysis.  
  
Extra credit options were added for computing *directed* degree correlation and clustering coefficient.

**Overview**

This assignment builds on your previous assignment, [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html). You will use your graph ADT implementation again (or you may borrow someone else's if yours didn't work well enough), and will implement various analyses of the graph structures.

* The assignment is designed to require that you take greater responsibility in algorithm design: you will need to figure out how to implement the analyses based on descriptions of requirements that give some guidance concerning the algorithms, but lack pseudocode. (The descriptions are by a physicist who is a respected expert in network analysis and gives primarily mathematical descriptions of requirements.)
* We will be comparing different types of networks that are studied in "network science" using graphs, including social, technological, and information networks. The analyses you conduct are chosen because they are known to discriminate between different kinds of networks, as well as because they make use of the graph ADT and one algorithm you learned for graphs, and will help you develop your skills in mapping requirements to an implementation.
* By making you the user of your own graph ADT code the project also helps you improve that code and understand how to write code that others will use. However, if your graph ADT was not working well enough to use for this project, you may use OSS, or use another class member's code (and they will get extra credit for that).

**Outline of Project**

1. **Reciprocity:** Compute the percentage of directed edges (*i*, *j*) for which there is a reciprocated edge (*j*, *i*). (Newman, 2010: section 7.10; you design the algorithm.)
2. **Degree Correlation** (a.k.a homophily, assortative mixing): Compute the correlation between degree(*u*) and degree(*v*) over all pairs for which there exists edge (*u*, *v*). (Newman, 2010: section 7.13, 8.7, 10.1). Regular credit for *undirected* degree, and extra credit for directed degree correlation (discussed later).
3. **Clustering Coefficient** (a.k.a. triadic closure): Compute the percentage of triads for which there exist edges (*u*, *v*) and (*u*, *w*) (the denominator) for which there also exists edge (*v*, *w*) (the numerator). Regular credit for *undirected* edges, and extra credit for directed clustering coefficients (discussed later). (Newman, 2010: section 7.9, 8.6, 10.2)
4. **Mean Geodesic Distance** (a.k.a. average path length): Compute the average length of the shortest path between each pair of vertices in the graph. While you are at it, also record the maximum non-infinite distance, or **diameter** of the graph. (Newman, 2010: sections 7.6 and 10.3)
5. **Analyze complexity of your solutions:** provide concise worst case analyses of the algrorithms you designed and implemented.
6. **Analyze a set of graphs using the above plus the analyses you did in Project 2.**
   * The graphs will include those from [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html) plus a few randomly generated graphs for comparison.
   * Your program will print the statistics from [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html) (SCC is optional), plus reciprocity, degree correlation, the clustering coefficient, and the mean geodesic distance.
   * Your report will summarize the apparent difference between social networks, information networks, and randomly generated networks.

**Motivations and Objectives**

Imagine that you are a recent ICS graduate and you have been hired at "rdIT", a local company for "research and development in Information Technology". Your job is to provide computational support to their research and development team. (rdIT is fictional, but similar companies exist in Hawaii, including Ikayzo, OceanIT, and Referentia.) Your boss may be smart and know a lot about the area in which you will be working, but obviously cannot do everything (else why would she have hired you?), so she expects you to handle the computer science aspects of the project. Thus as in any typical work situation you can expect to get a request that leaves a lot of details unspecified.

Your company is developing tools to support the analysis of networks (graphs) in a new area called "network science". (These tools already exist, but I've found that [an assignment requiring working at the cutting edge](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-3-2012.html) is a bit too much to ask at the ICS 311 level, so let's pretend that they don't.) Networks can be compared by various metrics, discussed in the next section. Network scientists are finding that some of these metrics may be useful for discriminating different kinds of natural and artificial networks. Many natural networks, such as information networks (e.g., citation patterns, WWW), technological networks (e.g., Internet router structure) and social networks (e.g., friend and follower relationships) differ from random networks, and these different kinds of natural networks also differ from each other. Your job is to implement the basic metrics and provide your boss with a preliminary report on what metrics seem to hold promise in discriminating the types of networks. (This latter question will be taken up in greater detail by your co-workers.)

**Metrics to Compute**

Part of the point of this project is for you to get practice figuring out how to develop an algorithm from a requirement, so I'm not saying a lot about the algorithms here.

**Reciprocity**

*Example: If I friend someone, do they friend me?*

Compute the percentage of directed edges (*i*, *j*) for which there is a reciprocated edge (*j*, *i*). See Newman (2010) section 7.10 for a description from which it should be easy to design the algorithm. You will just need to figure out the most efficient way to set up the iterations.

Reciprocity is higher in some kinds social networks than typical information or technological networks, because if someone establishes a relationship with you, you might feel obligated to do so in reply. However, this would not be the case when there are popularity disparities (e.g., if you follow a celebrity on twitter she is not likely to follow you).

**Degree Correlation**

*Example: Do people friend those of similar popularity?*

Compute the correlation between degree(*u*) and degree(*v*) over all pairs for which there exists edge (*u*, *v*). See Newman (2010) sections 7.13 for background, 8.7 for discussion of the formulas to be computed, and 10.1 for brief comments on run time. You will use formula 8.27, as discussed and broken down in that chapter. ***For regular credit, treat your graph as UNdirected.***

***For exra credit, treat your graph as directed.*** Use the methods in Foster et al. (PNAS 2010) "Edge direction and the structure of networks" (posted in Laulima) to compute the four degree correlation for these configurations: (in, in), (in, out), (out, in), (out, out). See end of this page for points (up to 25 points).

High degree correlation is a kind of *homophily*, or "love of same". The term *assortative mixing* is used to describe the level to which things of different types mix with each other. Human social structures may lead to higher homophily than other kinds of natural networks.

**Clustering Coefficient**

*Example: Are my friends friends with each other?*

Compute the percentage of triads for which there exist edges (*u*, *v*) and (*u*, *w*) (the denominator) for which there also exists edge (*v*, *w*) (the numerator). Newman (2010) discusses different ways to define clustering coefficient. Make sure you use the network-level metric, not Watts-Strogatz's local clustering coefficient. See section 7.9 for background, 8.6 for discussion of the potentially unpredictable complexity of this metric, and section 10.2 for further discussion on how to calculate the clustering coefficient and its complexity. ***For regular credit, treat your graph as UNdirected.***

***For exra credit, treat your graph as directed***. Use the methods in Fagiolo (arXiv 2007) "Clustering in Complex Directed Networks" (posted in Laulima) to compute clustering coefficients for these triangular patterns: cycle, middleman, in, out. See end of this page for points (up to 25 points).

The clustering coefficient is based on what is called *triadic closure* in some fields because it asks whether the partial triangle formed by (*u*, *v*) and (*u*, *w*) is closed by (*v*, *w*). Human social networks tend to have much higher clustering than other networks, especially random graphs. Hypothesized reasons include the introduction effect: I introduce my friends to each other; and the context effect: If I know two people by virtue of being in the same setting with them, then they probably know each other for the same reason (by being in the same setting).

**Mean and Maximum Geodesic Distance**

*Example: Is this a "small world"? Six degrees of Kevin Bacon?*

Mean geodesic distance is simply the *average path length* between pairs of vertices, and the maximum is known as the *diameter* of the graph. Compute the average length of the shortest path between each pair of vertices in the graph using shortest paths algorithms. Newman (2010, section 7.6) discusses some of the mathematical issues such as how to handle disconnected graphs, and Newman (2010, section 10.3) summarizes how to do it with iterated BFS or iterated Dijkstra's algorithm, both of which you have already studied. You may use either, but treat the graph as unweighted (all edges have weight 1) if you are using Dijkstra's, as the weights in the graph files mean something different here. (In most of these networks, higher weight means *stronger* connection, the opposite of *further distance*.) After searching from a given vertex, extract the numbers (path lengths for the numerator and number of pairs that have paths between them for the denominator) you need to compute the average (summing into accumulators), and also update the maximum if needed. Then you can throw away the tree before going on to the next source vertex: we don't need actual paths, nor do we need a matrix of all the distances.

Perhaps you have heard of "six degrees of Kevin Bacon", the claim that any actor is no more than six links away by co-appearance in movie links from Kevin Bacon; or the experiment by the psychologist Milgram that showed that it takes about six hops for a person to forward a letter to someone they don't know? In general, natural networks (not just social ones) tend to have surprisingly small diameters. This is the "small world effect", and it is common in *scale-free networks* that follow the *power law*. Some random graph models do not reflect this; other random graph models were developed specifically to model it better.

**Metrics from Project 2**

You will also compute these metrics using your code from [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html):

**Degree Distributions**

A summary of the minimum, average and maximum indegree and outdegree of the vertices graph.

**Graph Density**

Graph density is also used to characterize natural graphs. The density of edges affects whether we expect other phenomena, such as the likelihood that most vertices are in one giant connected component. Compute it as in [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html).

**Strongly Connected Components**

WCC and SCC are not as useful as community structure algorithms ([2012 version of Project 3](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-3-2012.html)), because they are too strict. Since it is complex and some of you may not have gotten it working, it is not required in this assignment, but if you got it working in [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html), include it in your output so we can see whether there are any interesting comparisons. The main point of interest is that scale free networks tend to have one very large component.

**Graphs for Analysis**

You will run algorithms on data we provide to report degree distributions and to find the strongly connected components.

**Data and Formats**

**Formats have not changed from**[**Project 2**](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html)**. This information is provided again here only for completeness.**

We are providing several graphs for testing, and may test your program on graphs different than the ones we provide.

The graphs are provided using Netdraw's VNA format, primarily because it is a simple format, easy to parse. See:<https://gephi.org/users/supported-graph-formats/netdraw-vna-format/>. Netdraw's .VNA format allows one to give vertices arbitrary names under the attribute ID, and to specify other attributes.

If you decide you want to make your ADT implementation a usable tool for future work, consider the more modern XML-based representations GEFX or GraphML: <https://gephi.org/users/supported-graph-formats/>. (These are already in the distribution directory: just change the extension of the URL.)

**.VNA**

The .VNA files list vertices after the line **\*Node data** or **\*Node properties** (there is a distinction between data and properties we won't be concerned with; case is not significant). Then edges are listed after the line **\*Tie data** (or \*tie data). A line of column headers follows each of these. After \*Node data there must be the column ID and there may be other data columns (see example below). After \*Tie datathere must be the columns **from to**, to indicate the source and target vertex, and there may be other columns, in particular **strength** for weighted graphs. Additional data on the vertices and edges may follow. You may ignore this additional data for this assignment, but it will be useful in interpreting the meaning of some of your results, so I suggest that you read the other attributes in as a string and store them in your graph.

Here is an example from the Gephi documentation cited above.

\*node data

ID name gender age

j101 joe male 56

w067 wendy female 23

b303 bill male 48

\*tie data

from to strength

j101 w067 1

w067 j101 2

j101 b303 1

w067 b303 5

Clearly, when you first read a line you expect \*node data followed by a line with the attributes. Then you read lines to create vertices, but check whether "\*" is the first character of each line. If it is, you expect it to be \*tie data, and then read in tie attributes and expect to read in ties until the file ends.

**No Simple Graph for Testing**

This assignment does NOT use the graph in [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html) constructed specifically to test the Graph ADT and SCC.

**Real World Networks**

You will test your program on the following graphs derived from real-world networks, and report the graph metrics from both Project 2 and Project 3. The networks are listed in order of increasing size. Use the best computer available to you (in terms of CPU speed, physical memory, and disk space available for virtual memory), and run it through the largest network you can get your software to work on. This is a test of the quality of your Graph ADT implementation!

**C. Elegans**: [**celegansneural.vna**](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/neural-c-elegans/celegansneural.vna)

A ***weighted, directed network*** representing the neural network of C. Elegans, a roundworm that is easily grouwn in the laboratory and widely used for biological research. The data were taken from the web site of Prof. Duncan Watts at Columbia University.   
***Source:*** [.http://cdg.columbia.edu/cdg/datasets](http://cdg.columbia.edu/cdg/datasets) (and <http://www-personal.umich.edu/~mejn/netdata/>)   
***Citations:*** J. G. White, E. Southgate, J. N. Thompson, and S. Brenner, "The structure of the nervous system of the nematode C. Elegans", *Phil. Trans. R. Soc. London* 314, 1-340 (1986). D. J. Watts and S. H. Strogatz, "Collective dynamics of `small-world' networks", *Nature* 393, 440-442 (1998).

------------------------------------------------------------

Graph celegansneural.net

------------------------------------------------------------

|V| = 297

|A| = 2345

Simple graph: TRUE

Density [ignoring loops] = 0.0266744

Density [loops allowed] = 0.02658459

Degree distribution: minimum average maximum

inDegree 0 7.895623 134

outDegree 0 7.895623 39

Total Degree 1 15.79125 134

Number of Weakly Connected Components: 1

Number of Strongly Connected Components: 57

Percent Vertices in Largest SCC: 80.47138

Reciprocity [ignoring loops]: 0.1680171

Reciprocity [loops allowed]: 0.1680171

Undirected Degree Correlation: -0.1520103

Directed Degree Correlation: -0.2327156

Clustering Coefficient [global]: 0.1807115

Clustering Coefficient [local]: 0.3079145

Mean Geodesic Distance: 3.991884

Undirected Diameter: 24

Directed Diameter: 35

Pajek Results that differ:

Watts-Strogatz Clustering Coefficient: 0.17814518

Network Clustering Coefficient (Transitivity): 0.10315218

Number of unreachable pairs: 20268

Average distance among reachable pairs: 3.99188

The most distant vertices: 263 (265) and 184 (240).

Distance is 14.

------------------------------------------------------------

**Political Blogs**: [political-blogs.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/political-blogs/political-blogs.vna)

Political blogosphere Feb. 2005. Data compiled by Lada Adamic and Natalie Glance. This is a ***directed*** graph according to who references whom, but ***unweighted*** (all the weights are 1.0). The original data classifies the nodes as "liberal" or conservative": this is coded as a numeric **value**. The node field **source** gives the name of the blog (there are duplicates). You may want to read in thevalue and source as node attributes to facilitate interpretation of results. Note: there are 3 self-loops.   
***Source:*** <http://www-personal.umich.edu/~mejn/netdata/>  
***Citation:*** Lada A. Adamic and Natalie Glance, "The political blogosphere and the 2004 US Election", in *Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem* (2005).

------------------------------------------------------------

Graph political-blogs.net

------------------------------------------------------------

|V| = 1490

|A| = 19025

Simple graph: FALSE

Density [ignoring loops] = 0.008575189

Density [loops allowed] = 0.008569434

Degree distribution: minimum average maximum

inDegree 0 12.76846 337

outDegree 0 12.76846 256

Total Degree 0 25.53691 467

Number of Weakly Connected Components: 268

Number of Strongly Connected Components: 688

Percent Vertices in Largest SCC: 53.22148

Reciprocity [ignoring loops]: 0.2425612

Reciprocity [loops allowed]: 0.2426807

Undirected Degree Correlation: -0.1960029

Directed Degree Correlation: -0.2307805

Clustering Coefficient [global]: 0.2259585

Clustering Coefficient [local]: 0.3600287

Mean Geodesic Distance: 3.390184

Undirected Diameter: 8

Directed Diameter: 9

Pajek results that differ:

Watts-Strogatz Clustering Coefficient: 0.23600779

Network Clustering Coefficient (Transitivity): 0.13843909

Number of unreachable pairs: 1237362

Average distance among reachable pairs: 3.39018

The most distant vertices: loveamericahatebush.com (380) and

americanworldview.tripod.com/weltansblog (794).

Distance is 9.

------------------------------------------------------------

**Wikipedia vote network**: [wiki-Vote.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/wiki-Vote/wiki-Vote.vna)

Wikipedia is a free encyclopedia written collaboratively by volunteers around the world. A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance. In order for a user to become an administrator a Request for Adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship. Using the complete dump of Wikipedia page edit history (from January 3 2008) Leskovec and colleagues extracted all administrator elections and vote history data. This results in 2,794 elections with 103,663 total votes and 7,066 users participating in the elections (either casting a vote or being voted on). Out of these, 1,235 elections resulted in a successful promotion, while 1,559 elections did not result in the promotion. About half of the votes in the dataset are by existing admins, while the other half comes from ordinary Wikipedia users. The network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent wikipedia users and a directed edge from node i to node j represents that user i voted on user j. 7115 Nodes, 103,689 Edges.   
***Source:*** <http://snap.stanford.edu/data/wiki-Vote.html>  
***Citations:*** J. Leskovec, D. Huttenlocher, J. Kleinberg. Signed Networks in Social Media. *CHI 2010.*; J. Leskovec, D. Huttenlocher, J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. *WWW 2010.*

------------------------------------------------------------

Graph wiki-Vote.net

------------------------------------------------------------

|V| = 7115

|A| = 103689

Simple graph: TRUE

Density [loops allowed] = 0.00204825

Density [ignoring loops] = 0.002048538

Degree distribution: minimum average maximum

inDegree 0 14.5733 457

outDegree 0 14.5733 893

Total Degree 1 29.14659 1167

Number of Weakly Connected Components: 24

Number of Strongly Connected Components: 5816

Percent Vertices in Largest SCC: 18.27126

Reciprocity [loops allowed]: 0.05645729

Reciprocity [ignoring loops]: 0.05645729

Undirected Degree Correlation: -0.06865903

Directed Degree Correlation: -0.08324456

Clustering Coefficient: 0.1254791

Clustering Coefficient [global]: 0.1254791

Clustering Coefficient [local]: 0.2088517

Mean Geodesic Distance: 3.341011

Undirected Diameter: 7

Directed Diameter: 10

Pajek results that differ:

Watts-Strogatz Clustering Coefficient: 0.11937507

Network Clustering Coefficient (Transitivity): 0.06818425

------------------------------------------------------------

**High-energy physics theory citation network**: [cit-HepTh.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/citation-HEPTheory/cit-HepTh.vna)

Arxiv HEP-TH (high energy physics theory) citation graph is from the electronic publication venue **arXiv** and covers all the citations within a dataset of 27,770 papers with 352,807 edges. If a paper i cites paper j, the graph contains a directed edge from i to j. If a paper cites, or is cited by, a paper outside the dataset, the graph does not contain any information about this. The data covers papers in the period from January 1993 to April 2003 (124 months). It begins within a few months of the inception of the arXiv, and thus represents essentially the complete history of its HEP-TH section. The data was originally released as a part of 2003 KDD Cup.   
***Source:*** <http://snap.stanford.edu/data/cit-HepTh.html>. (You can get paper authors, titles and abstracts here.)   
***Citations:*** J. Leskovec, J. Kleinberg and C. Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005. J. Gehrke, P. Ginsparg, J. M. Kleinberg. Overview of the 2003 KDD Cup. *SIGKDD Explorations* 5(2): 149-151, 2003.

**Enron email network**: [email-Enron.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/email-Enron/email-Enron.vna)

Enron email communication network covers all the email communication within a dataset of around half million emails. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. Nodes of the network are email addresses and if an address i sent at least one email to address j, the graph contains an undirected edge from i to j. There are 36,692 nodes and 183,831 edges. Note that non-Enron email addresses act as sinks and sources in the network as the dataset only includes communication with the Enron email addresses. The Enron email data was originally released by William Cohen at CMU.   
***Source:*** <http://snap.stanford.edu/data/email-Enron.html>. (The actual emails are available!)   
***Citations:*** J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 6(1) 29--123, 2009. B. Klimmt, Y. Yang. Introducing the Enron corpus. *CEAS conference*, 2004.

**Tapped In**: [ti-full.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/ti-full/ti-full.vna)

Graph of actors, chat rooms, threaded discussions, and files shared (collectively, "actants") in the TappedIn.org network of education professionals, Sept. 2005 through May 2007. There are just under 40K nodes and 200K edges. ***Directed:*** direction of arc indicates direction of influence, e.g., actor→room indicates that the actor chatted in the room and room→actor indicates that the actor listened to chat in the room. ***Weighted***: the weight indicates how many events took place between the actant nodes in the indicated direction. ***Bimodal***: edges only go between actors and artifacts (chats, discussions and files); never between actors and other actors, and never between artifacts. ***Multimodal***: You can infer the actant type from the node label: numbers are participants;ROOM indicates chat room; and DISC\_ID a discussion. The graph is very sparse, and will challenge any attempt at a matrix representation. It has thousands of SCCs but one is very large.   
***Citation:*** Suthers, D. D., & Chu, K.-H. (2012). Multi-mediated community structure in a socio-technical network. *Proceedings of LAK12: 2nd International Conference on Learning Analytics & Knowledge,* April 29 - May 2, 2012, Vancouver, BC. New York: ACM.

**Random Graphs**

The following "*G*(*n*,*m*)" random graphs have been defined, where *n* = |*V*| and *m* = |*E*|. The values of *n* and *m* have been chosen to correspond to the indicated real world networks. Both VNA and NET versions are available (change the extension on the URL for NET).

**G(297,2345)**: [g-297-2345.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/random/g-297-2345.vna)

For comparison to C. Elegans network celegansneural.vna. Average degree 7.896; Diameter 5; Density 0.02667440; Average path length 2.97677; Network clustering coeficient 0.02602816 (if you get 0.02613089 you are using the less preferred Watz-Strogatz method).

------------------------------------------------------------

Graph g-297-2345.net

------------------------------------------------------------

|V| = 297

|A| = 2345

Simple graph: TRUE

Density [ignoring loops] = 0.0266744

Density [loops allowed] = 0.02658459

Degree distribution: minimum average maximum

inDegree 2 7.895623 18

outDegree 2 7.895623 19

Total Degree 8 15.79125 30

Number of Weakly Connected Components: 1

Number of Strongly Connected Components: 1

Percent Vertices in Largest SCC: 100

Reciprocity [ignoring loops]: 0.02899787

Reciprocity [loops allowed]: 0.02899787

Undirected Degree Correlation: -0.05301033

Directed Degree Correlation: -0.02118495

Clustering Coefficient [global]: 0.05144291

Clustering Coefficient [local]: 0.05171182

Mean Geodesic Distance: 2.976772

Undirected Diameter: 4

Directed Diameter: 5

Pajek results that differ:

Watts-Strogatz Clustering Coefficient: 0.02613089

Network Clustering Coefficient (Transitivity): 0.02602816

The following results do not differ, but provide a clue: the methods

differ on how they treat unreachable pairs, as results are the same

when there are none:

Number of unreachable pairs: 0

Average distance among reachable pairs: 2.97677

The most distant vertices: v9 (9) and v252 (252). Distance is 5.

------------------------------------------------------------

**G(1490,19025)**: [g-1490-19025.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/random/g-1490-19025.vna)

For comparison to Political Blogs network political-blogs.vna. Average degree 12.768; Diameter 5; Density 0.00857519; Average path length 3.13612; Network clustering coefficient 0.00843601.

------------------------------------------------------------

Graph g-1490-19025.net

------------------------------------------------------------

|V| = 1490

|A| = 19025

Simple graph: TRUE

Density [ignoring loops] = 0.008575189

Density [loops allowed] = 0.008569434

Degree distribution: minimum average maximum

inDegree 2 12.76846 25

outDegree 2 12.76846 30

Total Degree 12 25.53691 46

Number of Weakly Connected Components: 1

Number of Strongly Connected Components: 1

Percent Vertices in Largest SCC: 100

Reciprocity [ignoring loops]: 0.007358739

Reciprocity [loops allowed]: 0.007358739

Undirected Degree Correlation: 0.01032546

Directed Degree Correlation: 0.01150463

Clustering Coefficient [global]: 0.01681392

Clustering Coefficient [local]: 0.01681722

Mean Geodesic Distance: 3.136125

Undirected Diameter: 4

Directed Diameter: 5

Pajek results that differ:

Watts-Strogatz Clustering Coefficient: 0.00843750

Network Clustering Coefficient (Transitivity): 0.00843601

------------------------------------------------------------

**G(7115,103689)**: [g-7115-103689.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/random/g-7115-103689.vna)

For comparison to Wikipedia vote network wiki-Vote.vna. Average degree 14.572; Diameter 6; Density 0.00204854; Average path length 3.61273; Network clustering coefficient 0.00197797.

------------------------------------------------------------

Graph g-7115-103689.net

------------------------------------------------------------

|V| = 7115

|A| = 103689

Simple graph: TRUE

Density [ignoring loops] = 0.002048538

Density [loops allowed] = 0.00204825

Degree distribution: minimum average maximum

inDegree 4 14.5733 30

outDegree 1 14.5733 36

Total Degree 11 29.14659 57

Number of Weakly Connected Components: 1

Number of Strongly Connected Components: 1

Percent Vertices in Largest SCC: 100

Reciprocity [ignoring loops]: 0.001755249

Reciprocity [loops allowed]: 0.001755249

Undirected Degree Correlation: 0.001811927

Directed Degree Correlation: -0.000808927

Clustering Coefficient: 0.003952244

Clustering Coefficient [global]: 0.003952244

Clustering Coefficient [local]: 0.003960416

Mean Geodesic Distance: 3.612734

Undirected Diameter: 4

Directed Diameter: 6

------------------------------------------------------------

**G(27770,352807)**: [g-27770-352807.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/random/g-27770-352807.vna)

For comparison to High-energy physics theory citation network cit-HepTh.vna. Average degree 25.40921858; Diameter 7; Density 0.00045751; Average path length 4.31034; Network clustering coefficient 0.00047644.

**G(36692,183831)**: [g-36692-183831.vna](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Network-Data/random/g-36692-183831.vna)

For comparison to Enron email network email-Enron.vna. Average degree 10.02022239; Diameter 13; Density 0.00013655; Average path length 6.70165 (large number of unreachable pairs); Network clustering coefficient 0.00014912.

**Tapped In?**

There is no random graph for comparison to the Tapped In network, as my random graph generation software does not generate bipartite multimodal graphs.

**Output Required**

You will print the summary statistics for each graph, extending the [Project 2](http://www2.hawaii.edu/~suthers/courses/ics311s14/Projects/Project-2.html) requirements with the additional ones for this project. You will not print the detailed SCCs.

**List of Statistics for All Graphs**

Print the following statistics for every graph you are able to load and analyze. These statistics help us verify that the graph was read in correctly, and also will be used to draw conclusions about the real world networks being modeled. The italicized ones are optional, and the bold faced ones are new to Project 3.

* number of vertices
* number of edges
* graph density
* minimum, average and maximum indegree
* minimum, average and maximum outdegree
* *number of strongly connected components (if you have it working)*
* *percentage of vertices in the largest strongly connected component (if you have it working)*
* **reciprocity**
* **degree correlation** − divided into (in, in), (in, out), (out, in), (out, out) if you are doing extra credit
* **clustering coefficient** − divided into cycle, middleman, in, out if you are doing extra credit
* **mean geodesic distance**
* **graph diameter**

**Output Format**

Please use this output format so it is easy for the TA to read. Replace ##### with the numbers. The metrics in italics are optional and those in bold are new.

------------------------------------------------------------

Graph *<filename>*

------------------------------------------------------------

|V| = #####

|A| = #####

Density = #####

Degree distribution: minimum average maximum

inDegree #### #### #####

outDegree #### #### #####

*Number of Strongly Connected Components: ####*

*Percent Vertices in Largest Strongly Connected Component: ####*

**Reciprocity: 0.###**

**Degree Correlation: 0.###**

**Clustering Coefficient: 0.###**

**Mean Geodesic Distance: ###.#**

**Diameter: ###**

Again, degree correlation and clustering coefficent will have 4 lines each if you do the extra credit extensions.

**Documentation and Discussion Required**

Requirements stated on the [Assignments Page](http://www2.hawaii.edu/~suthers/courses/ics311s14/Syllabus/Assignments.html) are included in the requirements for this project. Read that first. The following additional comments are specific to this assignment:

**Algorithms and Complexity**

In the **Reference Manual**, include a section for each of the four metrics you implemented (Reciprocity, Degree Correlation, Clustering Coefficient, and Mean Geodesic Distance & Diameter taken together). Each section should tell us where to find the code for each algorithm so we don't waste time searching for it. Also include pseudocode or a copy of the code directly in the reference manual to refer to in your analysis. Then each section should have a paragraph describing your algorithm, and another paragraph analyzing its run time complexity.

In the **Testing Document**, summarize the results in a table similar to that of Table 8.1 of Newman (2010) -- a column for each metric; a row for each graph. Then discuss the results briefly: do you see differences between information, technological, biological, social, and random networks? By what metrics: what metrics are potentially promising in telling them apart? There is a lot that could be said here: just say enough to show that you looked at the results and are able to draw some relevant conclusions. (Think of your boss as your audience: how much do you want to impress her? :-)

**Turning It In**

* **Readme.txt** is as specified in the page linked above. Be sure to include ALL instructions needed to compile your code, including other software it assumes is installed.
* **Operation Manual is not required** provided you cover the (simple) operation of this program in the Readme.txt.
* **Reference Manual** will include the algorithm descriptions and analysis specified above, along with other program design documentation specified in the [Assignments Page](http://www2.hawaii.edu/~suthers/courses/ics311s14/Syllabus/Assignments.html).
* **Testing Document** will include your results on all of the above graphs that your software could handle, comparison of results to those for which I have given known parameters (for checking correctness of your algorithms), and your discussion of what the metrics tell us about the networks.

**Grading**

**Points**

Some adjustments may be made when we see where the greatest effort is required. Total is 100 points. This assumes that you have run the analyses on the C.Elegans, Political Blogs, Wikipedia Vote, High Energy Physics, and Enron Email networks and on the five corresponding random graphs.

Program: 65 points

* 45 = 15x3 points for computing Reciprocity, Degree, and Clustering Coefficient.
* 20 points for computing Mean Geodesic Distance

Analysis and Documentation 35 points

These are in the Readme, Reference and Testing documents.

* 5 points for adequate Readme making it easy to run the program and briefly crediting use of others' OSS if applicable
* 20 points (5 points each) for adequate complexity analysis of your algorithms for computing the metrics, and documentation of their implementations, including what OSS you used (in the Reference Manual)
* 10 points for discussion of the results, comparing the different types of networks (in the Testing Manual)

**Extra Credit Points for Directed Graph Extensions**

Up to 25 points extra credit can be earned for each of directed degree correlation and clustering coefficient, in addition to the points already allocated::

* 15 additional points for adequate implementation.
* 5 additional points for adequate documentation in the Reference manual.
* 5 additional points for discussion of the results in the Testing document.

**Extra Credit Points for Being an ADT Provider**

If someone decides to use your graph ADT, you should ***both*** email ***both*** the instructor and the TA. (Also the person using your code should document this fact in their Readme.txt and Reference Manual.) As specified in the [Assessment](http://www2.hawaii.edu/~suthers/courses/ics311s14/Syllabus/Assessment.html)page, you will get up to 5 points for each "customer" in proportion to the extent to which they actually used your code, and to their grade.

Last modified: Sat May 3 05:05:57 HST 2014