Reference Manual

This application was written for use in Project 3 – Network Metrics for ICS311 Fall 2013 at the University of Hawaii at Manoa to understand graph theory and its algorithms and metrics.

The methods created were implemented in accordance with the interface given in the project specifications and influenced by Mark Newman’s ‘Networks: An introduction’.

[Javadocs](Javadoc/index.html) - information about specific methods and classes and their functionalities.

Analysis of each method

\*Data Structure used was an Adjacency List so methods that processes adjacent vertices are O(E)

Being that Project 3 is a continuation of Project 2. All methods used in Project 2 are used in this Project. Methods to calculate graph reciprocity, degree correlation, and clustering coefficient were added.

\*This document does not include results for **mean geodesic distance** and **graph diameter** metrics

// this method computes the fraction of reciprocal edges over all edges in // the graph

// for directed graphs edged *(u, v*) and *(v ,u)* form a cycle of length 2

// for undirected graphs all edges are reciprocal

// returns number of reciprocated edged / total number of edges

public double getReciprocity() {

return (double) edgecount / aList.numOfEdges

}

Runtime: Θ(V)

This method scans all out adjacent vertices and gets the undirected edge degree of each. Undirected because two directed edges that form a loop is equivalent to an undirected edge.

// this method computes the correlation between degree(*u*) and degree(*v*) over // all pairs which // there exists edge *(u, v).*

// a.k.a. graph assortativity or homophily

public double getDegreeCorrelation() {

return num / denom

}

Runtime: O(3V + E) = O(V+E)

// this method computes the fraction of paths of length two in the graph // which are closed over all // paths of length two in the graph

// a.k.a. the triadic closure

public double getClusteringCoefficient() {

return numberOfTriads / pathsOfLenghtTwo

}

Runtime: O(E^3)

[Return to Index](index.html)