

```

#include <cctype>
#include <cmath>
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include "console.h"
#include "filelib.h"
#include "grid.h"
#include "gwindow.h"
#include "simpio.h"
#include "pqueue.h"
#include "basicgraph.h"

using namespace std;

const string EGO_FILE_NAME = "348.edges";

/* Function: Load Ego Network
 * usage: loadEgoNetwork(graph);
 * -----
 * Loads the contents of an edge file into the ego graph.
 */
void loadEgoNetwork(BasicGraph & graph) {
    cout << "Loading facebook ego network... " << endl;
    ifstream fileStream;
    openFile(fileStream, EGO_FILE_NAME);
    string line;
    while(getline(fileStream, line)) {
        Vector<string> parts = stringSplit(line, " ");
        string a = parts[0];
        string b = parts[1];

        if (!graph.containsNode(a)) {
            graph.addNode(a);
        }
        if (!graph.containsNode(b)) {
            graph.addNode(b);
        }

        graph.addEdge(a,b);
        graph.addEdge(b,a);
    }
}

```

```

/* Function: Get Dispersion
 * usage: getDispersion(egoGraph, pointerToNode 475) -> 4996
 * -----
 * Calculates the dispersion score of a given vertex and an
 * ego network. See article: http://arxiv.org/pdf/1310.6753v1.pdf
 */
//
int getDispersion(BasicGraph & egoGraph, Vertex * amigo) {
    int count = 0;

    Set<Vertex *> mutualFriends = egoGraph.getNeighbors(amigo);
    for (Vertex * a : mutualFriends) {
        for (Vertex *b : mutualFriends) {
            if (a == b) {
                continue;
            }
            if (!egoGraph.containsEdge(a,b)) {
                count++;
            }
        }
    }

    return count;
}

int main() {
    setConsoleFont("courier-24");
    BasicGraph egoGraph;
    loadEgoNetwork(egoGraph);

    // should be 224 and 12,768 respectively
    cout << "Num nodes: " << egoGraph.getNodeSet().size() << endl;
    cout << "Num edges: " << egoGraph.getEdgeSet().size() << endl;

    // this should be 4996
    cout << "Dispersion of node 475: ";
    cout << getDispersion(egoGraph, egoGraph.getNode("475")) << endl;

    cout << "Calculating dispersion..." << endl;
    PriorityQueue<string> loverQueue;
    for(Vertex * v : egoGraph.getVertexSet()) {
        int d = -getDispersion(egoGraph, v);
        loverQueue.enqueue(v->name, d);
        if(loverQueue.size() % 10 == 0) {
            cout << loverQueue.size() << endl;
        }
    }

    cout << "The 10 most likely lovers:" << endl;
    for(int i = 0; i < 10; i++) {
        int priority = -loverQueue.peekPriority();
        string name = loverQueue.dequeue();
        cout << name << " " << priority << endl;
    }

    cout << "Thanks for playing" << endl;
    return 0;
}

```