DISCRETE PROJECT

By:

Aliza Ibrahim	21I-0470
Faiza Rehman	21I-0836
Kalsoom Tariq	21I-2487
Kissa Zahra	21I-0572

Table of Contents

Introduction:

In this project, we are given two datasets. One represents the Google web graph which has nodes representing the web pages and the edges representing the hyperlinks between them that directs you from one webpage to another. The other dataset represents the Twitter social circle, where the nodes represent profiles and the edges connecting them represent the social circles.

Problem Analysis

The goal of the project was to find the size of the Strongly Connected Components of both data sets.

We had options to choose between BFS and DFS to traverse through all the nodes and find the connections between them and then figure out how many strongly connected components there were. In the end we were to find the largest Strongly Connected Component and display its size.

Technique Applied

We decided to use the programming language C++ for this to make things a bit easier.

C++ comes with the STL library which makes the implementation of the code not only convenient but also efficient. We decided to apply the Kosaraju algorithm to implement the depth first technique to solve the required problem.

Implementation

Code Explanation:

```
19  class Undirected_Graph
20  {
21      // map object to know the status of node
22      map<int,bool>visited;
23      // map object containing Adjacency list
24      map<int,list<int>> Adjacency_list;
25
26      // Performs DFS and fills the stack in a given order
27      void FillStack_inOrder(int , stack<int>&);
28      // A recursive function to find the size of each SCC
29      int SCC_Size(int&,int);
30
31      // Returns an object containing Transpose of graph
32      Undirected_Graph Transpose();
33
34  public:
35
36       //The Function to be called from main to invoke code
37        void Size_of_LargestSCC();
38
39      // Adding Edges in Graph between nodes
40      void addEdge(int,int);
41
42  };
```

Firstly, we made a class by the name of Undirected graph. This will be the container to store our adjacency lists for each vertex and to mark them as visited and unvisited as we begin the traversal. As you can see, there are four main functions we have employed the use of.
An Undirected Graph object is created in the main. First an adjacency list is created for all the nodes using the addEdge(int,int) function called by the created object. Then we call the function Size_of_LargestSCC(). This function is what starts the entire process. It employs Kosaraju's algorithm to find the strongly connected components.
Then we use the SCC_Size(int&,int) function to find the size of the largest component in the given dataset and print it.

Output analysis:

The console prints these values for the given data sets separately as shown above. In the case of the google dataset, the largest number of web pages connected with each other through hyperlinks is 434818.

In the case of the twitter dataset, the largest number of profiles connected with each other in the form of a social circle is 68413.