CSE 5400 Interdisciplinary CS

Assignment 1:WHO IS THE MOST CONNECTED

Instructor:Dr.Chan

Student name:Lingfeng Zhang

ID:9017490960

In the class, we talked about how to figure out the importance of each vertex in a graph,while the length of all the edges in the graph are same.We came up with three algorithms, and used two of them to implement.

The algorithm 2 is called degree of separation, which is a classical BFS algorithm, aims to find out the all the shortest paths from each vertex to the rest of the vertexes and calculate the average length. The algorithm 3 is called betweenness, which is based on the algorithm 2, but the goal is to find out which vertex appears the most times in all the shortest paths.

A. Beside the “friendship”,what other relationship can be represented by an edge in social network?

   1. The cell phone communication system. Since the speed of Hertz wave is so fast, we can regard the time which is cost between every two land-stations is same.So the vertexes are stations and the edges are the exist channel between two stations.

2. The flight system. Each airport only has the limited air routes to some other airports. If we only care about how many shortest times we need to transfer from the start to the destination, the flight system is similar to the “friendship”.The vertexes are airport and the edges are air routes.
3. The maze graph. In a maze graph,there are many intersections, and people will choose a direction at each intersection in order to find out the exit.So we can regard the intersections as vertexes and the path between two intersections as edge.Then we figure out the path from the start position to the exit.

B. Compare the two algorithms.

1. Quality of the output

For the example toy-bowtie, using algorithm2 can get the top 3 vertexes are d,c,e(from more important to less important). Using algorithm3 can get the top 3 vertexes are d,c,e(from more important to less important). So using the two algorithms can get the same output.

For the example toy-friends, using algorithm2 can get the top 3 vertexes are Jobs,Gates,Disney(from more important to less important). Using algorithm3 can get the top 3 vertexes are Gates,Disney,Jobs(from more important to less important). Even

through the top 3 are the same,but they are not in the same order.

For the example toy-graph, using algorithm2 can get the top 3 vertexes are 21,22,12(from more important to less important). Using algorithm3 can get the top 3 vertexes are 21,12,22(from more important to less important). Even through the top 3 are the same,but they are not in the same order.

So that means a vertex in a graph may has a smallest degree of separation, but it may not appear the most times in all the shortest pairs.

2. Time/speed

While using the algorithm 2, we suppose that there are V vertexes and E edges, and for each vertex V_i there are E_i edges connected with it. For algorithm2,because we use the adjacency list,we need to check every E_i for every V_i ,so all the edges will be visited. So for visiting the vertexes it will be cost O($\sum_i^n Vi$)=O(V), and for visiting all the edges it will be cost O($\sum_i Ei$)=O(E). In a graph,the number of edges is no more than n(n-1)/2,n represents the number of V, so in the worst situation the O(E)=O[(n^2-n)/2].So the time complexity for algorithm2 to find out the single source is O[(n^2+n)/2]=O(n^2).

For algorithm 3, the basic steps depend on the algorithm2,using the adjacency list to collect each vertex.But the difference is to create a table to save the paths from the start vertex to the present

vertex.Because I use the recursion to deal with the present “parents” vertexes and their “children” vertexes.For the worst situation the recursion function will be repeated (n-1) times,n is the number of vertexes. And in each time, the “children” vertexes will inherit their parents paths, it will cost C_i*B_i steps, C_i represents the length of the path and B_i represents the number of ways. So the time will be cost $\sum_i^{n-1} Ci * Bi$,so the time complexity in the worst situation will be $O(n^2)$ for the single source.

3. Space/memory

For algorithm 2, I use the queue structure to save the vertexes.Using n to represent the number of vertexes,and the structure is 12 byte, so the Space complexity is O(n*12)=O(n)

For algorithm 3, I also use the queue structure to save the vertexes,and I add a matrix to save the shortest paths and the passing notes.In a graph,n represents the number of vertexes,the max number of the shortest paths between two vertexes is n-2, and the max length of the shortest paths between two vertexes is n-1.So the space of matrix is (n-1)*(n-2), and there are n vertexes in the queue,finally the Space complexity for algorithm 3 is $O[n*(n-1)*(n-2)]=O(n^3)$.