# Graph Mining Lab 3: Documentation

**Aayush Keval Shah**
**2019A7PS0137H**

Q**1.** In the given question, we need to generate an orientation based upon (1) degree of the vertices (2) graph degeneracy (3) number of triangles contained per vertex. For breaking a tie, use the vertex id and Report the maximum indegree and maximum outdegree as the output of your implementation for graph-1 and graph-2.

We first find all the degrees of the nodes in the graph. We then compare the degrees of the neighbouring vertices. We then make the edge directional based on the degrees (from low degree to higher). Then we find the max indegree and outdegree based on the current graph. We then continue the same step till the graph is directional. Then we print out the max indegree and outdegree. We do the similar thing with number of triangles per vertex and graph degeneracy also.

Output for graph1 :
Using vertex degree, The maximum indegree is 19 The maximum outdegree is 7
Using the number of triangles contained per vertex, the maximum indegree is 19 and maximum out degree is 13
Using graph degeneracy the maximum indegree is 11 and maximum out degree is 19

Q**2.** In the given question, we need to implement an algorithm for counting all cliques of size k in the input graph from the paper "Arboricity and Subgraph Counting" by Chiba and Nishizeki and implement an algorithm for counting all maximal cliques from the same paper.

First take a node with a degree equal to that of the input k. Then find another node with the same degree or more and find the number of cliques Then again take another node and repeat the same until all k cliques are found,recurse this algorithm for different nodes

Output of graph2:(when k=5)

Reading edgelist from file edgelist.txt
Number of nodes = 334864
Number of edges = 925872
Building the graph structure
max degree = 6
Number of nodes = 334864
Number of edges = 925872
Number of 5-cliques: 61551

Q**3.** In the given question, we need to implement the Community Detection algorithm due to Girvan & Newman discussed in the class.

The algorithm's steps for community detection are summarized below

1. The betweenness of all existing edges in the network is calculated first.
2. The edge(s) with the highest betweenness are removed.
3. The betweenness of all edges affected by the removal is recalculated.
4. Steps 2 and 3 are repeated until no edges remain.

Output:

Q**4.** In the given question, we need to implement the implement Node classification using the label propagation technique discussed in the class.

The process has 5 steps:[2]

1. Initialize the labels at all nodes in the network. For a given node x, $C_x$ (0) = x.

2. Set t = 1.

3. Arrange the nodes in the network in a random order and set it to X.

4. For each $x \in X$ chosen in that specific order, let $C_x(t) = f(C_{xi1}(t), ...,C_{xim}(t),C_{xi(m+1)}(t-1), ...,C_{xik}(t-1))$. Here returns the label occurring with the highest frequency among neighbours. Select a label at random if there are multiple highest frequency labels.

5. If every node has a label that the maximum number of their neighbours have, then stop the algorithm. Else, set t = t + 1 and go to (3)