COL-380 : Assignment 3

Again Viral Marketing

Deadline: April 3, 2023

This assignment will be an extension to the assignment 2. The dataset and underlying problem setting will remain same as in Assignment 2.

1 Tasks

The assignment is focused on combining shared-memory (OpenMP-based) and messsage-passing (MPI-based) programming. Both Task 1 and Task 2 are required of COL380 students. There is no additional task for COV880. You may not select a new partner. If you want to resign from your group and do this assignment individually, please email col380ta.

- **Task 1: Advertisement** This is same as assignment 2. You are finding groups – edge-induced subgraphs.
- **Task 2: Ego-Network** This is same as assignment 2.

  Note that **not** all of the vertices of a group need to have an edge to an influencer vertex. The influencer vertex simply has to be connected to at least one vertex in a group. An influencer vertex can be either part of a group or it must be one of the immediate neighbour of the vertices in these groups. A vertex is an immediate neighbour of another vertex if there is an edge between both.

2 Deliverable

A zip archive with the filename *<FirstName>_<EntryNo>_A3.zip*. The format for the entry number should be *20XXCSXXXXX*.

On unzip, it should produce the following directory structure:

```
<FirstName>_<EntryNo>_A3/
    |-- Makefile
    |-- report.pdf
    |-- metric.csv
    |-- other code files
    |-- executable
```

– The following command needs to be executed to run your program:
  – *make*: This should create an executable for your program named as a3.

- For task 1 the output format is the same as the assignment 2.
- For task 2, there are some changes

- In case verbose = 0, setting, if there is no influencer vertex then the output should contain a single value i.e 0. If there are any then first output the number of influencer vertices and on the next line output influencer vertices in any order.
- In case verbose = 1, if there exist c (s.t. $c > 0$) influencer vertices, then first output the number of influencer vertices. On the next 2c lines, for each influencer vertex(in any order), first output the influencer vertex, and on the next line output all the vertices of the groups influenced by this influencer vertex, space-separated and can be in any order. If there is no influencer vertex then simply output 0 on a new line.

Use all available memory on each process.
In the final report you should run your tasks with np up to 8 (i.e 2, 4, 8), and for each node, you can consider cores up to 8 (at least 4 and 8).

Provide a comprehensive report, including graphs and tables for speedup and efficiency. Describe in detail the approach you took for each task and describe your observations. Estimate the sequential fraction of your solution for 8-64 cores. Indicate how well your code scales (try to estimate iso-efficiency, and explain how you did it).

Stick to the provided report format. In final submission along with *report.pdf* you need to submit a *csv* file named as *metric.csv*, with the provided format(find here). You can download the template csv file from ***File > Download > (.csv)***. To automatically fill in the metric score, you can optionally choose to use the Python package pandas.

```
Report Format
    |- Detailed Approach for Task 1
    |- Plot Graph for speedup and efficiency for different core counts
    |- Report Iso-efficiency and explain
    |- Estimate sequential fraction for 8-64 cores
    |- Justify why your solution for task 1 is scalable
    |- Detailed Approach for Task 2
    |- Plot Graph for speedup and efficiency for different core counts
    |- Justify why your solution for task 2 is scalable
    |- References
```