COL 761 - Homework 2

- You need to do the homework in your already formed team of 3. Make sure to upload your code to the GitHub repo you mentioned in HW0.
- Due: **30th September 11:59PM.**
- Your code must compile and execute on HPC. You may get 0 for compilation or execution errors
- **Do not copy code from your friend or from the internet. Any plagiarized code will result in an F grade for the course.**

Submission Instructions:

- Submit a script (**<RollNo>_install.sh), on Moodle (do not zip it). Only one per team. Multiple submissions will be penalized.** Like previous assignment, it should contain one line to clone your repository, one line to cd into it, and load module commands (only. Nothing else, don't unzip the archive.) Provide github's https link, not the ssh one.
- Upload HW2_<RollNo>.zip file to your GitHub repository. This RollNo should be of one of your team members. Ex. HW2_CSZ198347.zip. On unzipping it should **produce one folder** of the same **name as the zip file.** This folder should contain
  - **all the source code files**,
  - a **README.txt** (i) explains all the files you have bundled, (ii) has entry numbers and names of **all team members** (iii) has instructions on how to execute your code and (iv) contribution in percentage of each student.
    There should be minimal manual overhead required to run your code.
  - a **report file** containing all explanations with plots named as <rollnumber.pdf> eg. CSZ198347.pdf .

- Make sure to use the same roll number to give the name of all submission files (HW2_<RollNo>.zip , RollNo.sh), etc.
- Do not hardcode the dataset and query file names in your code for question 2.
- There should be **only one submission** per group (on github and moodle).
- Latest timestamp of your last pushed commit and Moodle submission of install.sh will be treated as your submission time.
- Marks will be distributed to individual members based on their percentage contribution. For example, if someone has contributed 20%, then his/her marks will be 20/33*(marks obtained. However, no extra marks will be given for contributing more than 33%. In case of disputes, we will go by majority vote within that team.)

- Please use only the modules available on HPC, do not install any new packages as HPC doesn't allow this. (Using –U flag to install python packages in user mode is also discouraged as it installs a package which can clash with other people's code that we'll be evaluating after yours.)

1. This question is about getting yourself familiar with frequent subgraph mining tools. Run it on the Yeast (https://bit.ly/3EVA5Cc) Dataset. This is a database of molecules. The format is the following:

```
#graphID
# of nodes
Series of Node Labels
# of edges
Series of "Source node, Destination Node, Edge label"
```

Run gSpan, FSG (also known as PAFI), and Gaston (you should be able to find it online) against frequency threshold in the dataset (you may need to write a script to change the format of the dataset) at minSup = **5%, 10%, 25%, 50%** and **95%**. Plot the running times and explain the trend observed in the running times. Specifically comment on the growth rates and why one technique is faster than the others. You are free to consult the respective papers. **(20 points)**

Libraries you can potentially use:

- gSpan : https://sites.cs.ucsb.edu/~xyan/software/gSpan.htm
- FSG : http://glaros.dtc.umn.edu/gkhome/pafi/download
- Gaston : https://liacs.leidenuniv.nl/~nijssensgr/gaston/download.html

2. Consider the problem of subgraph search. You are given a database of graphs $D = \{G_1, \cdots, G_n\}$(Ex. Chemical compounds), and a query subgraph Q. Your goal is to identify all data graphs that contain (subgraph isomorphism) the query subgraph. As we know subgraph isomorphism is NP-hard. We need a better method than brute force. So, here is an idea of an index structure. Mine m subgraph patterns from D. Convert every data graph G_i into a binary feature vector $F_i = [f_{i,1}, \cdots, f_{i,m}]$ of dimension m such that $f_{i,j}$ indicates whether pattern j is contained within G_i or not. Now, given Q, you can construct its feature representation as well. It is now easy to see that Q can be subgraph isomorphic to some data graph G_i only if all features contained in Q are also contained in G_i. Thus, you can prune out all data graphs that do not satisfy this criteria and perform subgraph isomorphism only on the remaining graphs. Several questions however remain unanswered. What should be the value of m? The number of frequent subgraphs is likely to be very high. How do you prune it down to m? What should be the frequency threshold? Your task is to solve these questions so that the time for subgraph isomorphism is as small as possible. You need to submit the following
   - A script that will take a file containing a set of graphs (maximum size of graph database is 100,000). The script should build the index structure first. Maximum time allowed to built this structure is 10 minutes (sh index.sh <graph dataset>).
   - Once the index is built, it should ask for another file as input containing the query graphs (sh query.sh <queries_file.txt>).
   - For each query, you need to provide the IDs of all graphs containing that query. If there are n graphs in the query file, the output file should have n lines; each line containing the IDs in tab delimited manner. The name of the output file should be output_**<RollNo>**.txt.

- Assumptions: You may assume that all graphs will be chemical compounds. However, we will evaluate you on datasets other than Yeast. The dataset may contain up to 50k graphs.
- You can use the VF3 library for subgraph isomorphism tests (MiviaLab/vf3lib: VF3 Algorithm - The fastest algorithm to solve subgraph isomorphism on large and dense graphs (github.com)).

Marks

a. Correctness of results. 30 points*Avg. Fscore across 30 queries.
b. Efficiency: Avg. of (Fastest running time/Your time)*30. You will get 0 for queries that do not return the correct result. This does not include the indexing time.

3. Use elbow plot to determine the correct value of k in k-means clustering on the dataset generated by generateDataset.sh. [20 points]

Instructions to generate dataset for elbow plot question:

1. Unload any existing module:
   `$ module purge`
2. Load the following module on HPC:
   `$ module load compiler/gcc/9.1.0`
3. Run the file `dataset_gen_hpc_compiled` (provided on Moodle) to generate dataset as:
   `$ ./dataset_gen_hpc_compiled CSZ198347`
4. Make sure the file has execute permissions (running `ls -l` should give `-rwxr-xr-x` if not use command `chmod ug+x dataset_gen_hpc_compiled`).
5. This should generate a file named `generated_dataset.dat`
6. Different roll numbers will give different datasets, so all teammates use the one that will be used to name the submission file.
7. Use the correct format: CSZ198347 is **correct**. 2019CSZ8347 is not. Also, use capital letters. csz198347 is also not correct.