# Assignment 3: PageRank, Frequent Item-Sets and Clustering

Formative, Weight (15%), Learning objectives $(1, 2, 3)$,
Abstraction (4), Design (4), Communication (4), Data (5), Programming (5)

**Due date:** $11:59$ **pm,** 9 **May,** 2022

## 1 Overview (Attention, Different To Previous Assignments)

This assignment must be done **individually**. This means all the rules regarding individual submission will apply and the submission must be solely your own work. Therefore, we will not use the groups on MyUni. You will need to submit on the assignment page as an individual.

## 2 Assignment

**Exercise 1** *Frequent Item-Sets (30 points)* ***(Postgraduate Students Only (COMP SCI 7306))***

Suppose there are 100 items, numbered 1 to 100, and also 100 baskets, also numbered 1 to 100. Item i is in basket b if and only if i divides b with no remainder. Thus, item 1 is in all the baskets, item 2 is in all fifty of the even-numbered baskets, and so on. Basket 12 consists of items 1, 2, 3, 4, 6, 12, since these are all the integers that divide 12. Answer the following questions:

1. If the support threshold is 5, which items are frequent?
2. what is the confidence of the following association rules?
   (a) $\{5, 7\} \rightarrow 2$.
   (b) $\{2, 3, 4\} \rightarrow 5$.

**Exercise 2** *PageRank (40 points)*

1. Implement the PageRank Algorithm as discussed in Section 5.1 and 5.2 (Leskovec, Rajaraman and Ullman) in JAVA, Python or C++. Your implementation should make use of the improvements regarding efficiency and the methods of dealing with dead-ends and spider traps. There are several PageRank implementations available on the web. You have to do your own implementation without using any code from other sources.

2. Run your algorithm on the Google Web Graph 2002 available at

   http://snap.stanford.edu/data/web-Google.html

   and provide a file listing the PageRank for each node. Report separately, the ordered list of the ten nodes having the largest PageRank

Your approach should be efficient as possible in terms of runtime and memory requirements.

Note: you are asked to implement the algorithm from scratch, without using third party implementations/ libraries.

**Exercise 3** *Clustering (30 points)*

1. Perform a hierarchical clustering on the one-dimensional set of points and show your results (best to use dendrograms)

   $1, 4, 9, 16, 25, 36, 49, 64, 81.$

   assuming the clusters are represented by their centroid (average), and at each step the clusters with the closest centroids are merged. (Exercise 7.2.1)

2. Implement the K-means algorithm and carry out experiments on the Iris dataset (note that you are not allowed to use the libraries such as scikit-learn to implement the algorithm itself, but you are free to compare your results with such). The dataset can be accessed from scikit-learn library. You may follow the instructions at the following link:

   https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

   a) Plot the K-means clustering results by plotting the first 2 dimensions of the input data as well as the converged centroids.

   b) Provide some discussions about how you picked the value of K in the K-means algorithm.

   Note: You should only use the 4 input **features** in the Iris dataset to cluster them, and not the **labels**. Also, similar to previous exercise, you are asked to implement from scratch without using third-party implementations/ libraries.

## 3 General assignment submission guidelines

As stated in the beginning of the assignment, work MUST be submited using the group's interface on MyUni, and a single submission per group, ONLY. The submissions will include the following, at minimum:

- PDF file of your solutions for theoretical assignments. The solutions should contain detailed description of how to obtain the result.
- All source files, all the project files.
- PDF or txt file with descriptions of your implementations to understand your code.
- Files containing the results of your algorithms on the provided datasets.
- PDF or txt file of your computation times of the algorithms on provided datasets.
- a README.txt file containing instructions to run the code, student ID, and email address.
- the submissions that do not follow the above guidlines may lose points accordingly.

Please do not hesitate to reach out using the discussion forum, workshops, or the contact details of the teaching assistants on the home page of MyUni, should you have any questions or concerns.