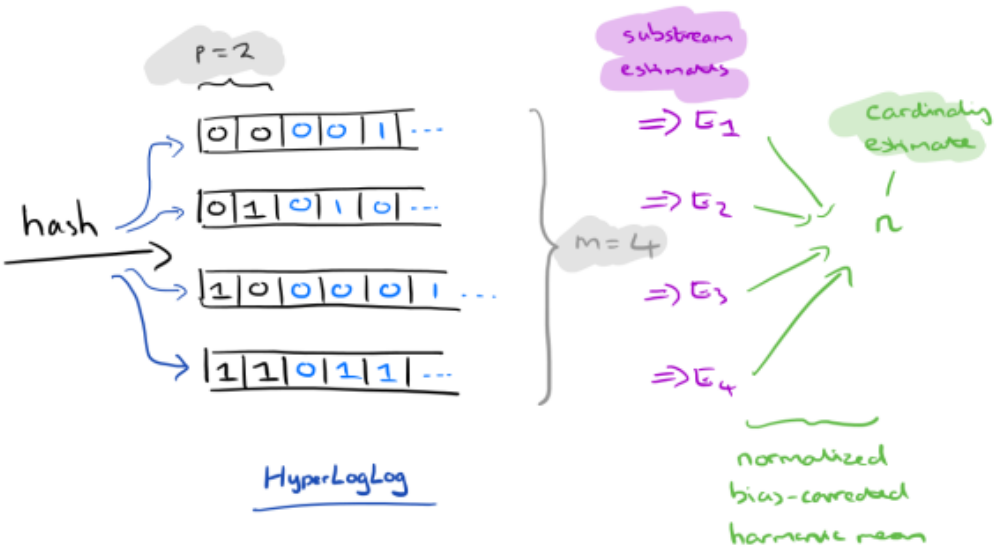
Homework 4 - Hard coding

Goal of the homework: write important algorithms and functions from scratch.

1. Hashing

For this task, we are dealing with hashing algorithms. In particular you will implement **hash functions** and an algorithm called ***HyperLogLog*** (HLL).

There are many scenarios where we need to know (or at least estimate) the cardinality of a dataset, e.g., statistical purposes. Consider the need to determine the number of distinct people visiting a website. For famous social media (e.g., Facebook, Instagram) or e-commerce sites (e.g., Amazon, ASOS), this task could be computationally expensive because of the large number of users. Doing this with traditional methods (e.g., SQL query) on a dataset as massive as the sites mentioned above could take days and large amounts and memory. Is it not a better idea to approximately estimate the number of distinct users? HyperLogLog is an algorithm that allows us to make decent guesses when counting vast numbers of distinct elements, with very little computation or memory required.



Your task

Download the datasets here. You will find a `hash.txt` file.

1. Implement your hash functions by scratch, *no ready-made hash functions are allowed*. Read the class material and search the internet if you need to. As a

reference, it may be useful to look at the description of hash functions in the book or here.
2. Use your hash function, implement a HyperLogLog structure.
3. Read the dataset sequentially and add it to your HyperLogLog.
4. At the end *you have to provide*:
    - The cardinality of the dataset.
    - The error of your filter.

2. Clustering

We play with a dataset gathering reviews (~560k) of fine foods from Amazon. The reviews include much information. We focus on the reviews' plain text and try to cluster the products (~74k).

To solve this task, you must:

1. Implement the k-means clustering algorithm (**not** ++: random initialization). We ask you to write the algorithm from scratch following what you learned in class.
2. Run the algorithm on the food data. Then, use the already implemented version of k-means++, are there any differences in results?
3. Analyse the obtained clusters:
    - Identify the kind of products in the cluster (e.g., chips, tea, coffee) using a visualization called word cloud.
    - Provide the number of product in each cluster
    - Compute the reviews' score distribution in each cluster. Once you get them, test if their mean differences are statistically significant!
    - Get the number of unique users writing reviews in each cluster

Before running the algorithm, you should consider the following:

- How do you represent the data? (e.g., do I use a binary representation or TF-IDF?)
- How do you pre-process data? Since you aim to characterize products by their review, do you want to consider words that appear in too many or too few documents?
- After organizing your data, you will realize that tens of thousands of words compose your vocabulary. In this case, we suggest you to use the SVD method to reduce the dimensionality of the dataset. This operation is typically used to denoise the data and implies the loss of some information. For this reason, you first reduce your dataset to a few hundred components (e.g., 100). Then, you use the following `np.cumsum(explained_variance_ratio_)` to verify the total amount of variance you retain with an increasing number of components. You can pick a

number of components that retain> 60% of the variance. Since we know this point can raise many questions, opening a thread on Slack is recommended and welcomed.
- The choice of the number of clusters should **not** be random.

IMPORTANT: We are aware that you may consult the internet for information about implementing the requested algorithms. However, the final code must be yours! So please, do not search and copy-paste the code.

Bonus

We remind you that we consider and grade the bonuses only if you complete the entire assignment.

1. Implement the k-means algorithm using MapReduce.
2. Dynamically visualize the clustering evolution after each iteration (use a sample of the product belonging to each cluster to do it!)

3. Algorithmic question

You are given an array with A with n integer numbers.

- Let s = min{ A[1], ..., A[n] } and b = max { A[1], ..., A[n] }.
- Let r = b - s

Prove that we can sort A in time O(n + r).