# Locality Sensitive Hashing (LSH)

Instructor: Kunpeng Zhang (kzhang6@uic.edu)
TA: Minghong Xu (mxu29@uic.edu)

This assignment is optional. You can receive 10 extra credit points if you finish it by May 1 without any bugs. (No partial credits)

The input of your project would be a folder with many textual documents. The output would be clusters where similar documents are together. Given the similarity threshold s, you need to tune your parameters r (the size of each band), b (the number of bands), and the number of min-Hashing functions based on the probability that at least one band is identical:

$$1-(1-s^r)^b$$

To be specific, each line in your output file must be in the following format:
clusterID: docID, docID, docID, …, docID

Three important implementation steps:

**1) Convert documents into big vector matrix.**

Notes: Before singling, please do some preprocessing, such as stop word removal, stemming, etc. Use individual word as singling token and the token size of 1 are reasonable in practice. The big vector matrix is actually a term-document occurrence matrix. We do not consider the number of appearance times for each word, because of the Jaccard similarity function. After this step, you should have one or multiple files like:

$$doc_i, term_1, term_5, \ldots, term_{104}, \ldots$$

where the list of terms above occur in the $i^{th}$ document $doc_i$.

**2) Implement min-Hashing to get signature matrix.**

Notes: one-pass implementation

For each column C and the $i^{th}$ hash function h(i) keep a slot for the min-hash value
Initialization: sig(C)[i] = ∞
Scan rows looking for 1s
Suppose row j has 1 in column C
Then for each h(i):
If $k_i(j) <$ sig(c)[i], then sig(c)[i] ← $k_i(j)$
// where $k_i(j)$ is the $j^{th}$ row number in permuted order

After this step, you should have a signature matrix M.

**3) Hash bands to cluster similar candidates.**

Notes: Create a hash bucket (B) with the size of k (as large as possible). Split your signature matrix M into b bands and each band has r rows. Design your own hash function to hash

each band $b_{ij}$ (the $i^{th}$ band for the $j^{th}$ document) into one of the bins in bucket B. If two documents appear at least one time in the same bin, then they are candidate pairs.

**Requirements:**

- Please comment important parts of your codes to make more readable.
- When you submit your codes through blackboard, you need to put all source codes (.java files, NOT jar files), at least 100 text files in a folder, and some other optional files (e.g., a readme file) into one folder and name that folder as <YOUR UID>_ASSIGN6. Assignments not following this rule will not be graded. In addition, no resubmission after TA grades it. Late submission rule: 10% deduction for one day late. Late submission over a week is NOT acceptable.

DO NOT copy any codes from others. Otherwise, both will be penalized.