**Report** **107062546 楊仲愷**

**Java File, Class Name: Shingle.jar, Shingle**

**Minhashing.jar, Minhashing**

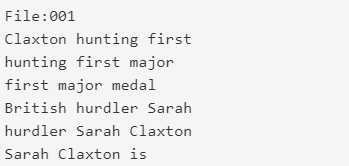
**LSH.jar, LSH**

**Similarity.jar, Similarity**

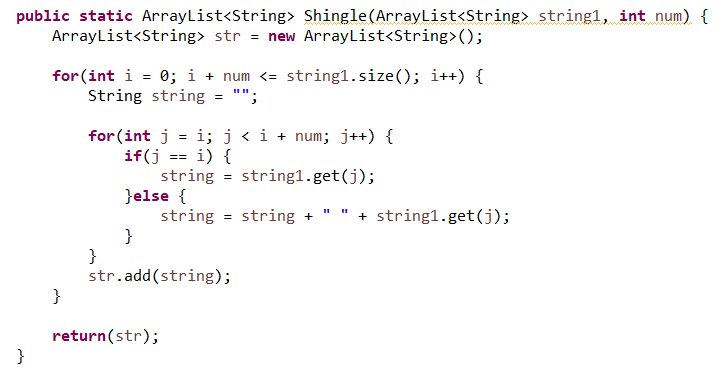
First, please execute Shingle.jar, and we will get the shingle file. Total input files are in the path, /hw02/Input, and the output path we set /hw02/Output001.



Then, we will get all the files which are processed into 3-shingle.



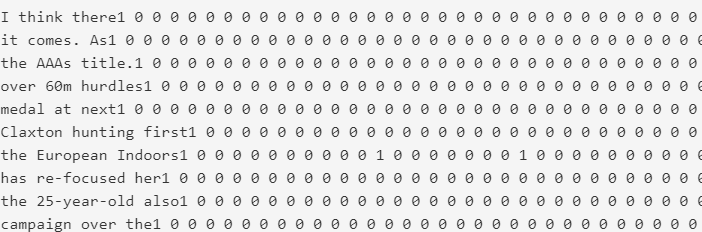
The function shingle will divide string into several parts, and num can be input by the shingles we want.



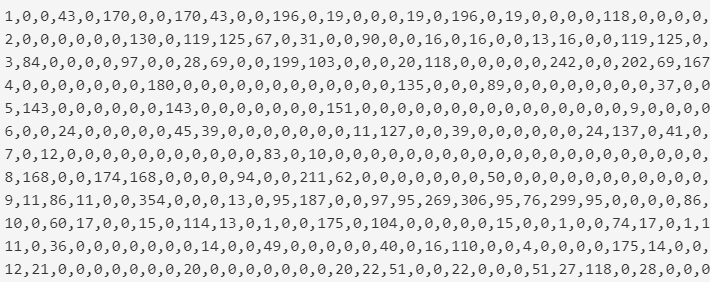
Second, we take advantage of minhashing to check the relations with different shingled file with permuted by different hash functions.



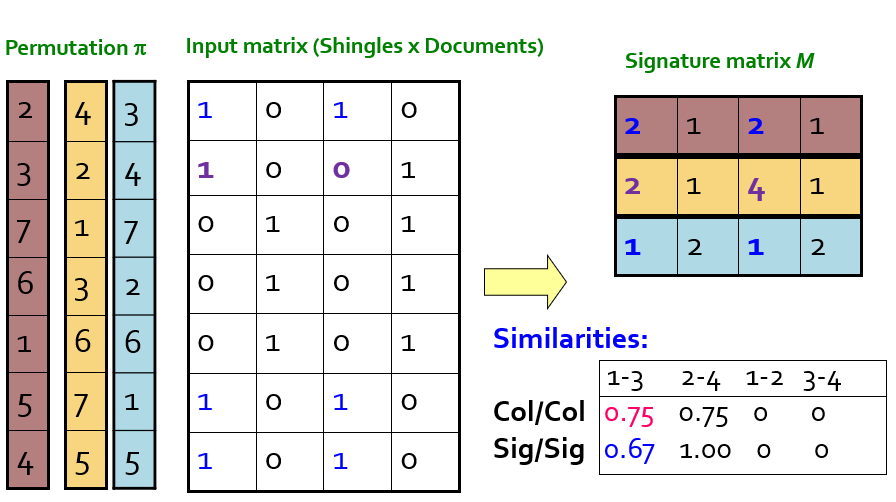
We will get files which are processed with permuted shingles. And 1 means the document contains the shingle.



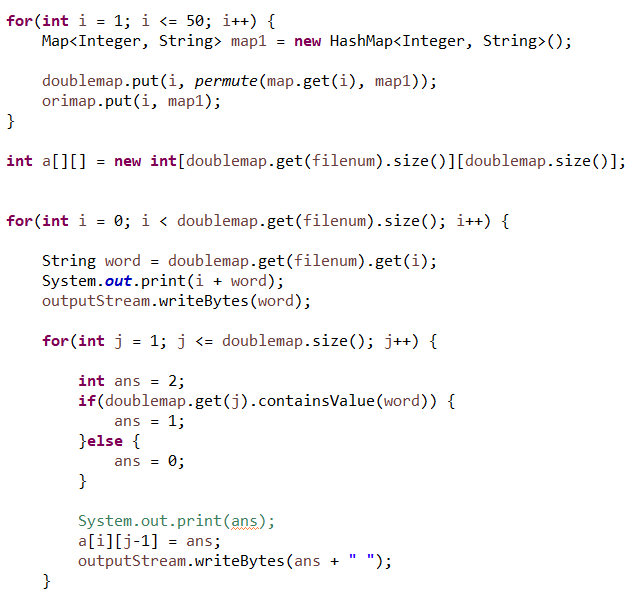
And we also get a file named total.txt processed by minhashing. It is the signature matrix.

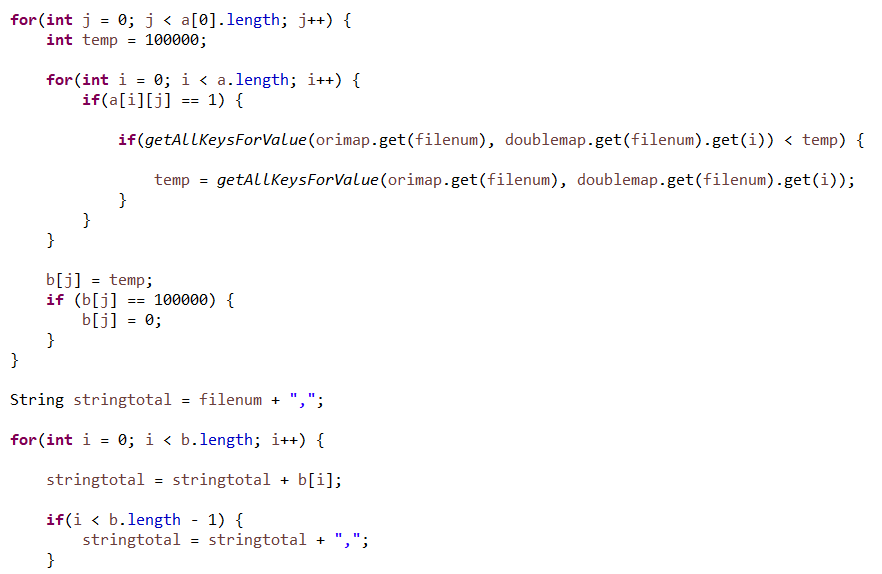


Minhashing:

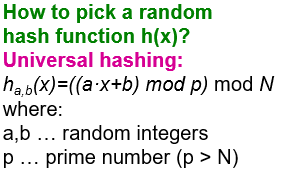


I permute every shingle file and find every column which contains the shingle.

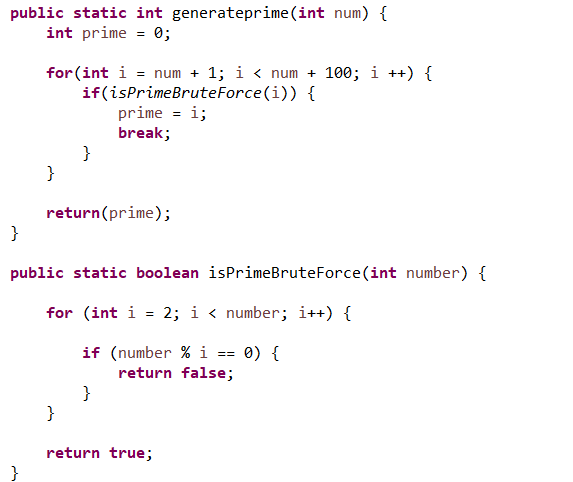


Then, I build the matrix and find the smallest number of shingle and represent

The way I generate different hash functions.



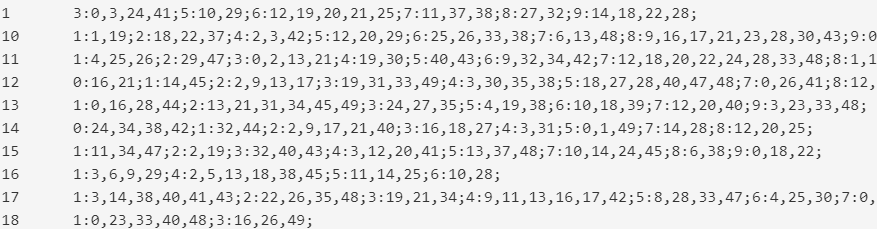
The way I generate different prime numbers of hash functions. I search the next 100 numbers of input and determine whether it is a prime number or not with the function isPrimeBruteForce.



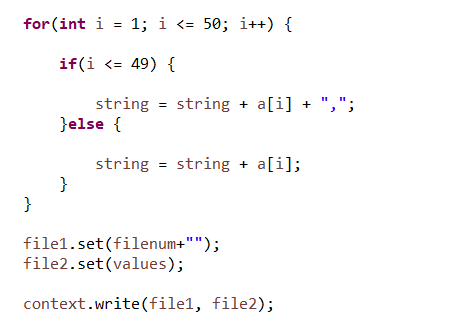
Let’s go to the third part, locality sensitivity hashing. We use the results which processed from the second part.



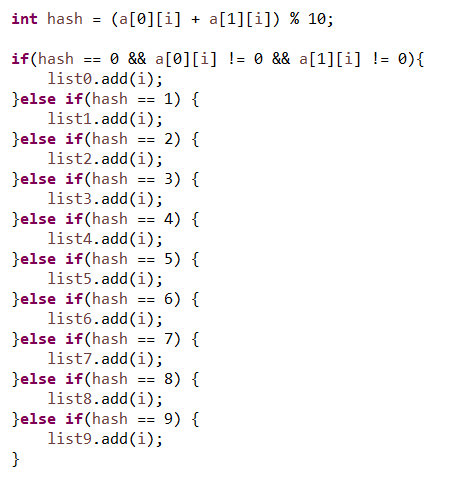
Then, we put the candidate pairs into the hash function and show the buckets which contain over 2 files.



I set the band number to key and set the row of signature matrix to value.



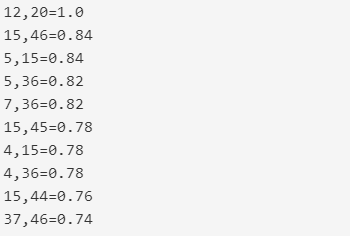
The locality sensitivity hashing I used is I add the two of rows in a band and moderate 10 into different buckets.



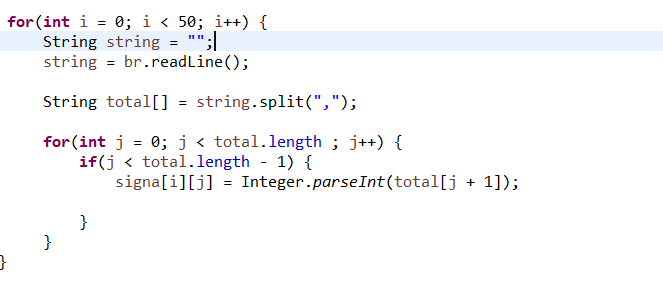
Finally, we execute Similarity



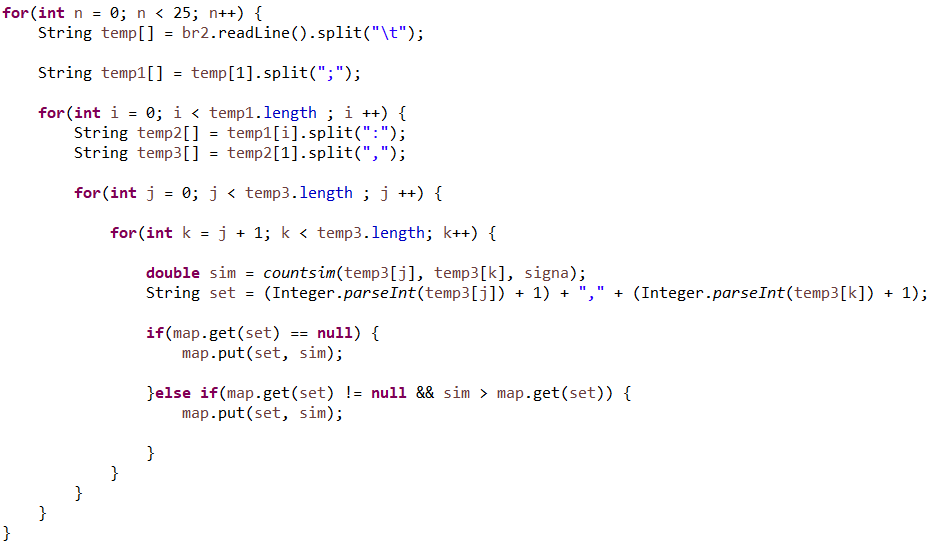
We will get the answer.txt which represent the top 10 candidate pairs and similarities.



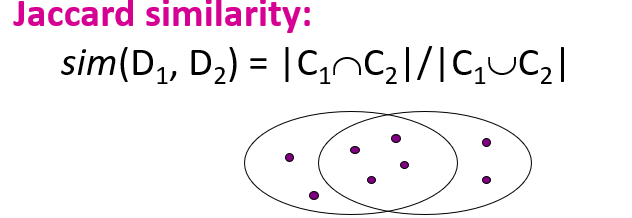
We get the signature matrix firstly.

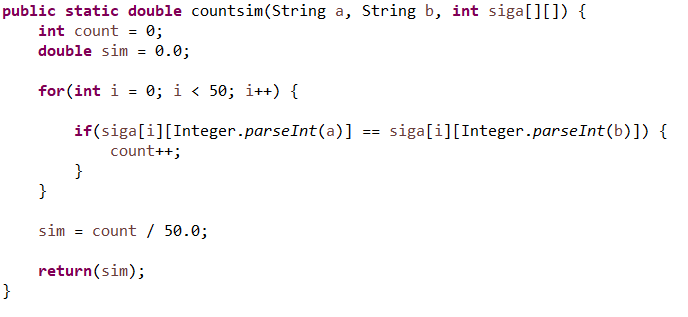


Then, we count the similarities of candidate pairs and get the highest one.



This is how we calculate similarities. I get the one set of candidate pairs and get the signature matrix of the two columns and use Jaccard similarity.





And sorted by value; get the top 10 pairs and similarities.

