**Report** **107062546 楊仲愷**

**Java File, Class Name: FrequentItemsets01, FrequentItemsets**

**FrequentItemsets02, FrequentItemsets**

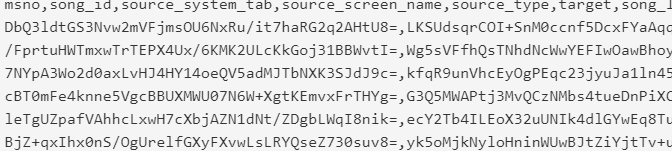
**FrequentItemsets03, FrequentItemsets**

**I implement frequent itemset algorithm and take advantage of a priori algorithm. Data are from the log of kkobx users.**

In jar FrequentItemsets01, first, I deal with the data from total2.txt, and I will get the processed data which are songs listened by users and output them and named output-01.txt because we can drop out the data which we don’t use, and the size of file can be smaller.



Total2.txt:



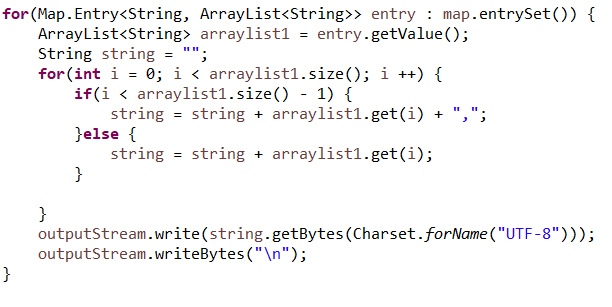
output-01.txt:



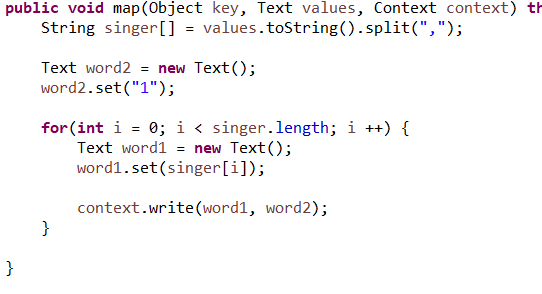
Read total2.txt:



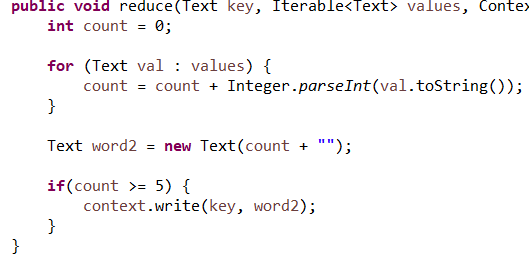
Write output-01.txt



Then, we count the number of songs which are listened. In mapper, we set the name of song with key and 1 with value.



In reducer, we count the number of song and only write songs which number are higher than 5.



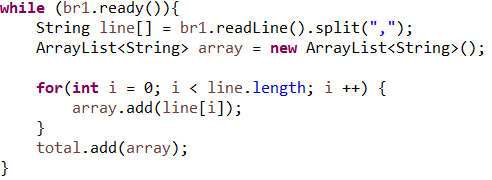
And we will get the output.



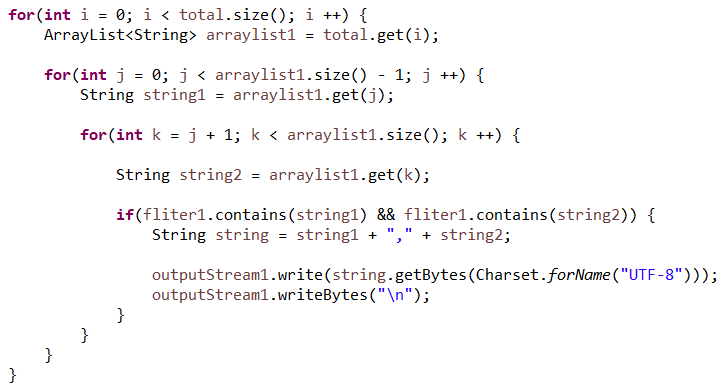
In jar FrequentItemsets02, we read the output from FrequentItemsets01, take advantage of a priori algorithm, and only write the data which are not deleted; then, write into freq2.txt. And count the number from freq2.txt.



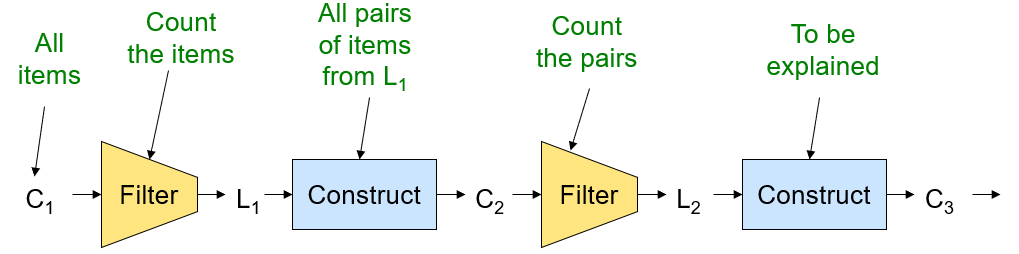
Read output01.txt

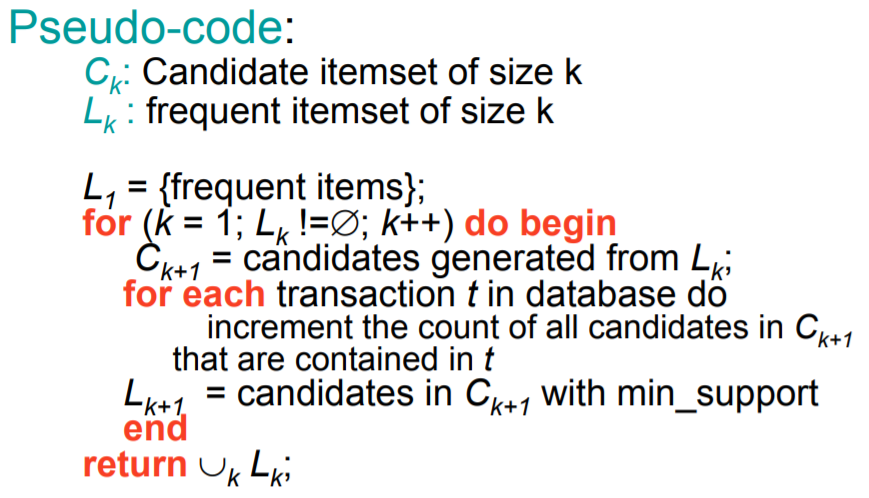


Write the data which are not deleted:



A Priori Algorithm:





freq2.txt:

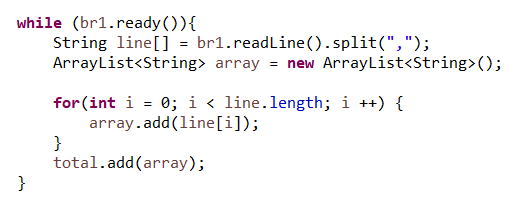


And the mapper and reducer are like FrequentItemsets01. We will get the output.

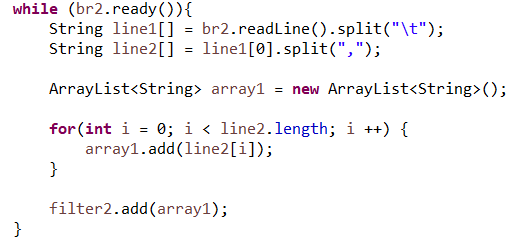


In jar FrequentItemsets03, we read the output from FrequentItemsets02, take advantage of a priori algorithm, and only write the data which are not deleted; then, write into freq3.txt. And count the number from freq3.txt.

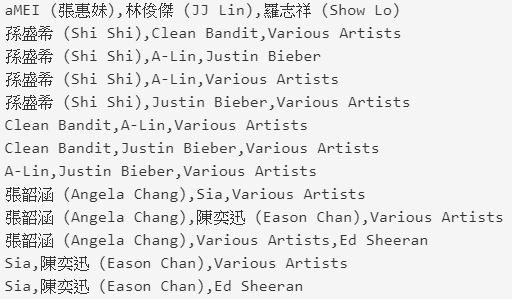
Read the output from the original listened songs:



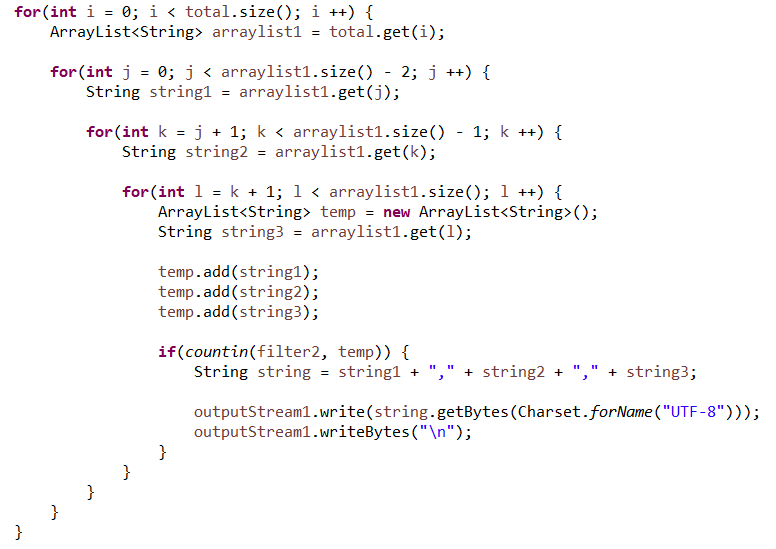
Read the output from FrequentItemsets02:



freq3.txt:



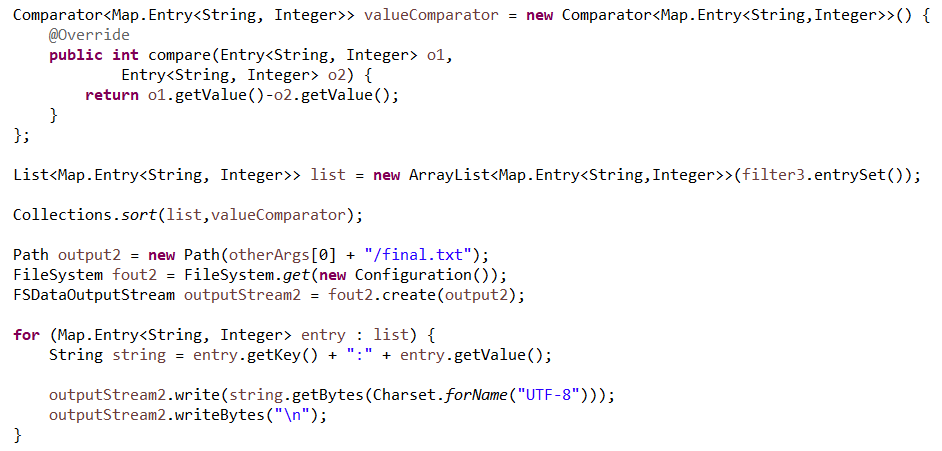
Here, we take advantage of an algorithm which are from myself. I read every two-tuple set and if each of them is in three-tuple set and count every item in three tuple set and counts will be higher than two.



output:



In the final, I sort the output frequent itemset by their counts and get final.txt.



final.txt:

