COMP9313 Big Data Management

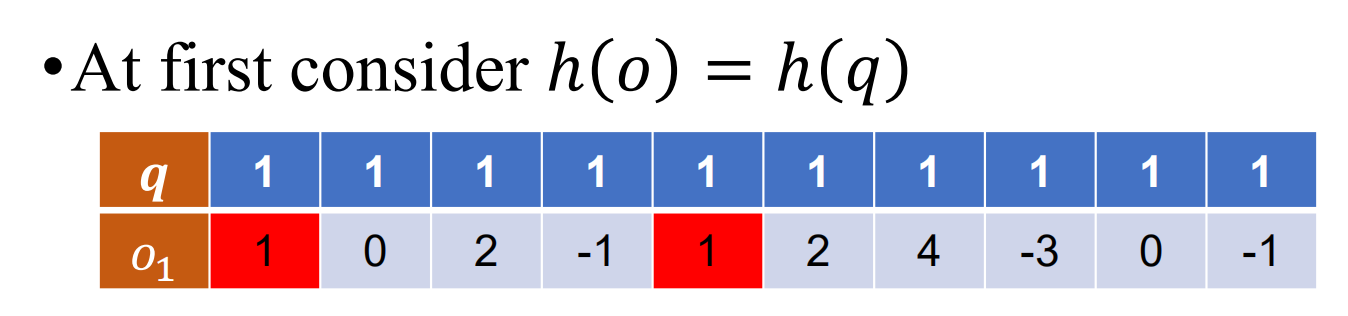
Project 1 Report(Term2 2020)

Student Name: Zhenming Wang Student ID: z5140192

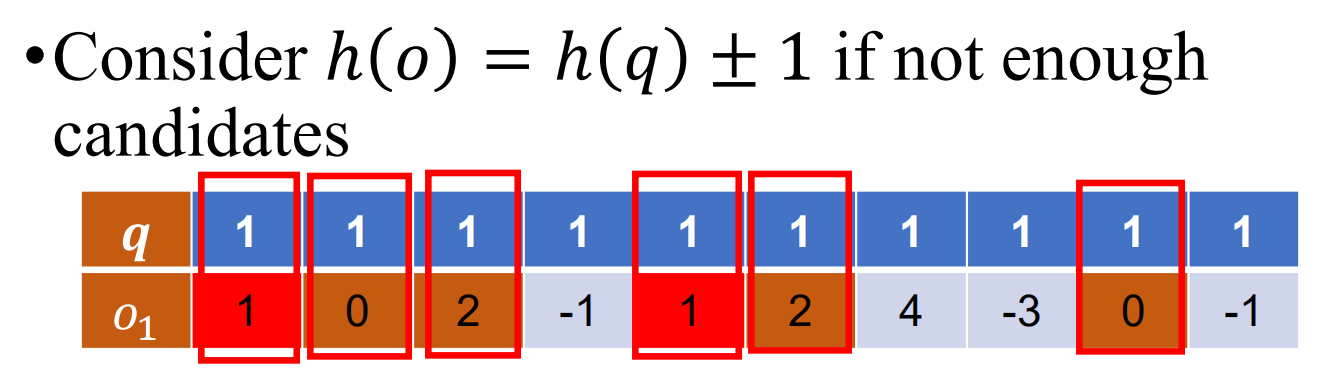
Part1: Understanding of Virtual Rehashing

This part of the knowledge is the basis for completing the project, so first show the bottom-level knowledge principle

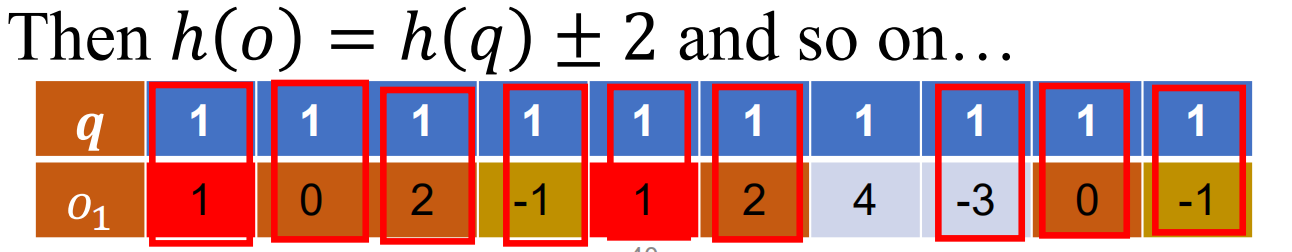
Set the Collision number = 8



Only two number can match, the Collision number=2<8, So it need to relaxing conditions.

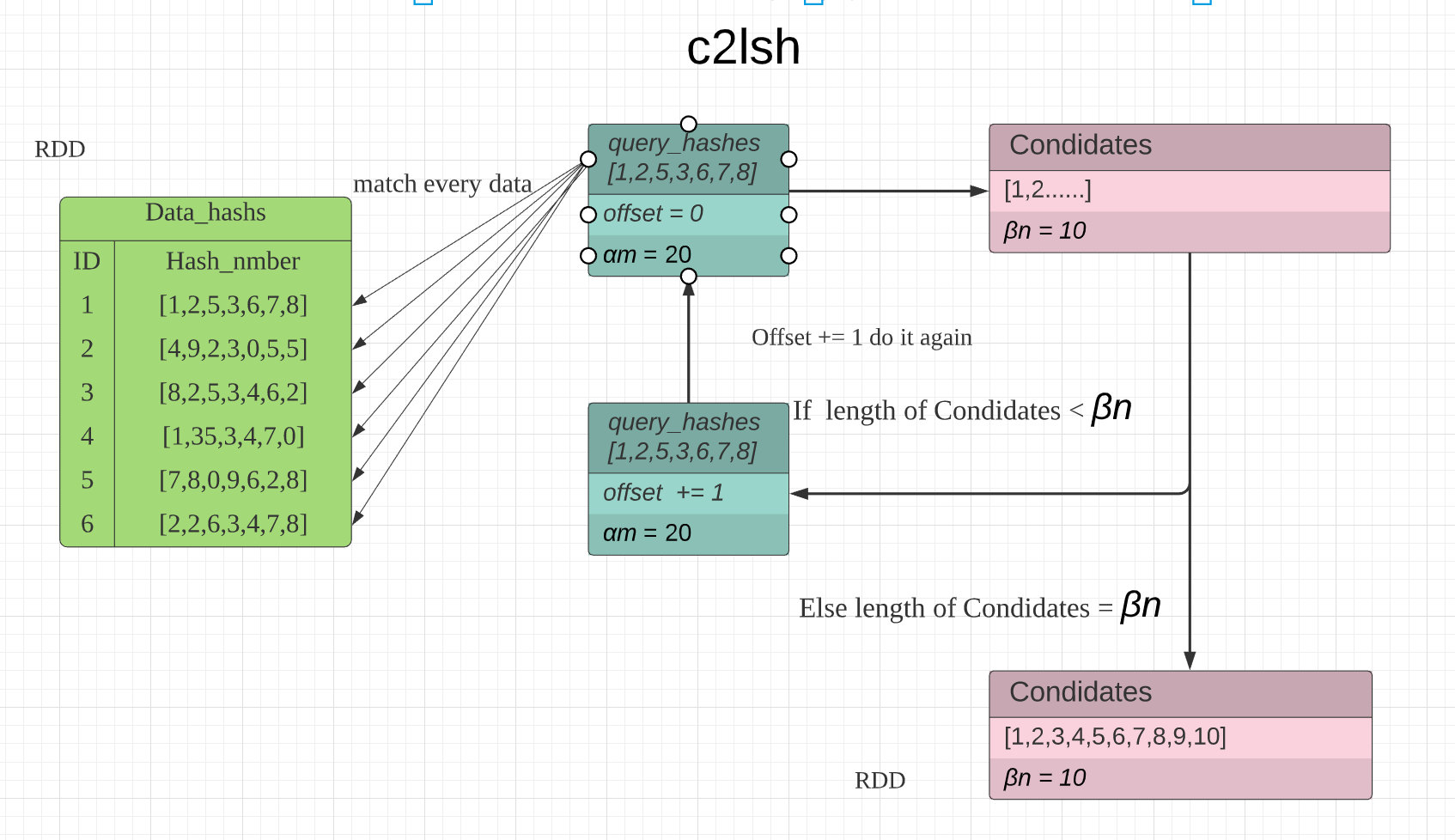


The Collision number=6<8, So it also need to relaxing conditions.



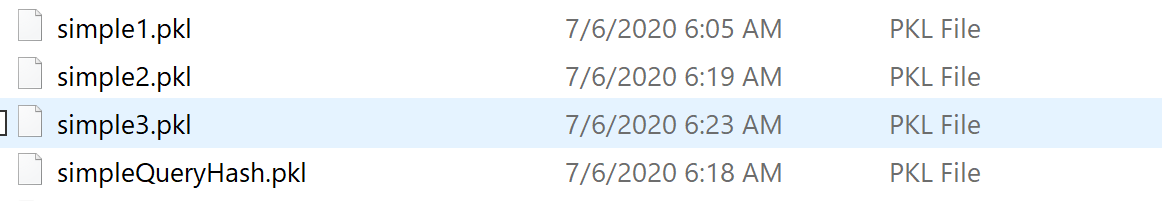
Finally, Collision number=8 and Complete match,the task completed

Part 2: Implementation details of c2lsh()

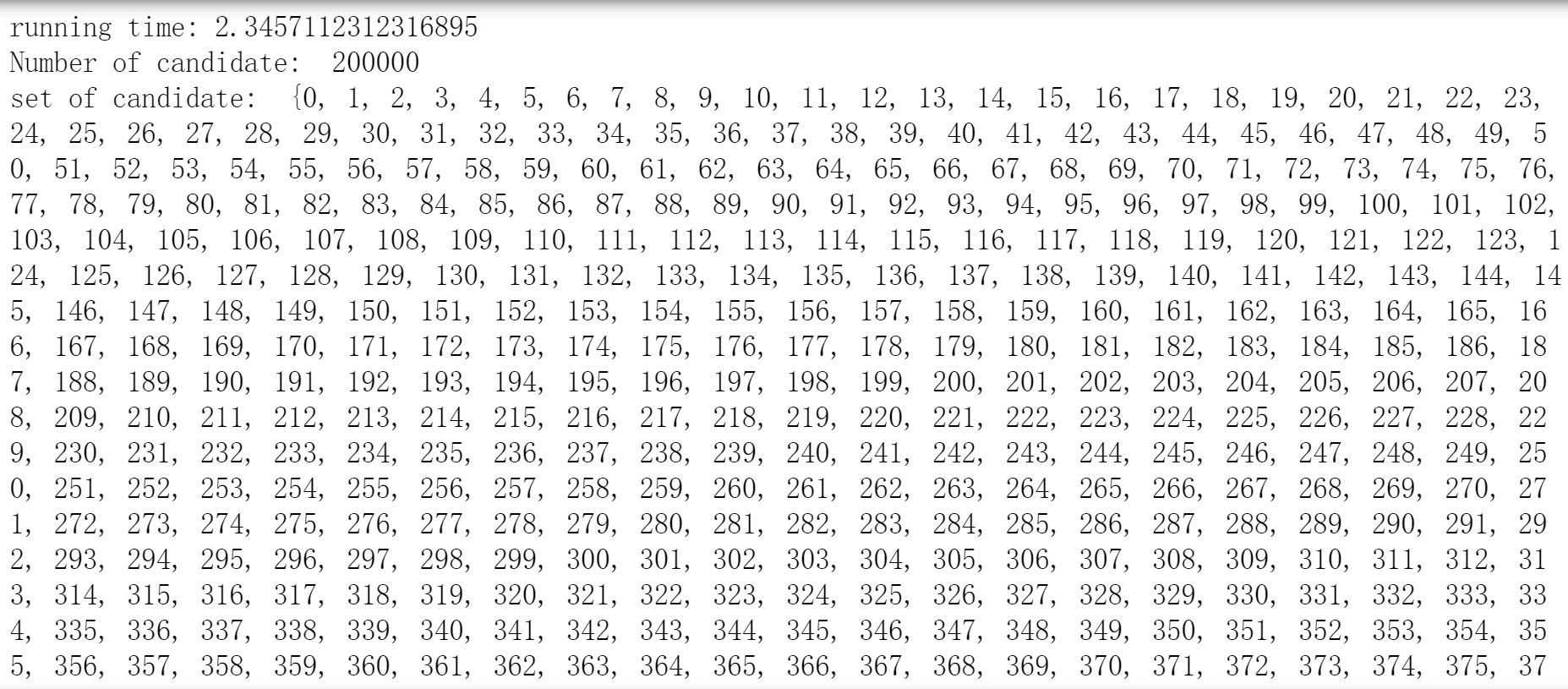


Part 3: Show the evaluation result of my implementation using my own test cases.

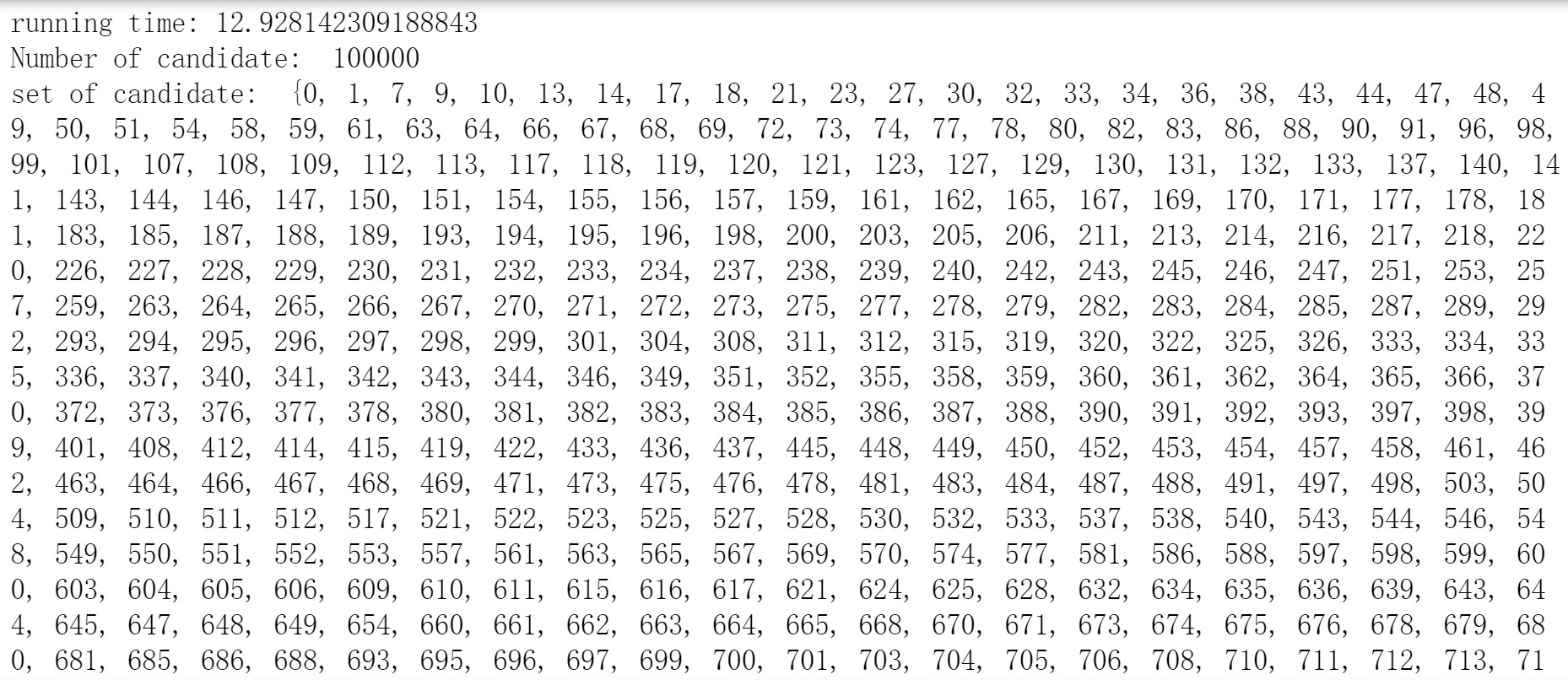
I randomly created three test sets, the simple1 size is 2000000, the simple2 size is 400,000, and the simple3 size is 1000000.



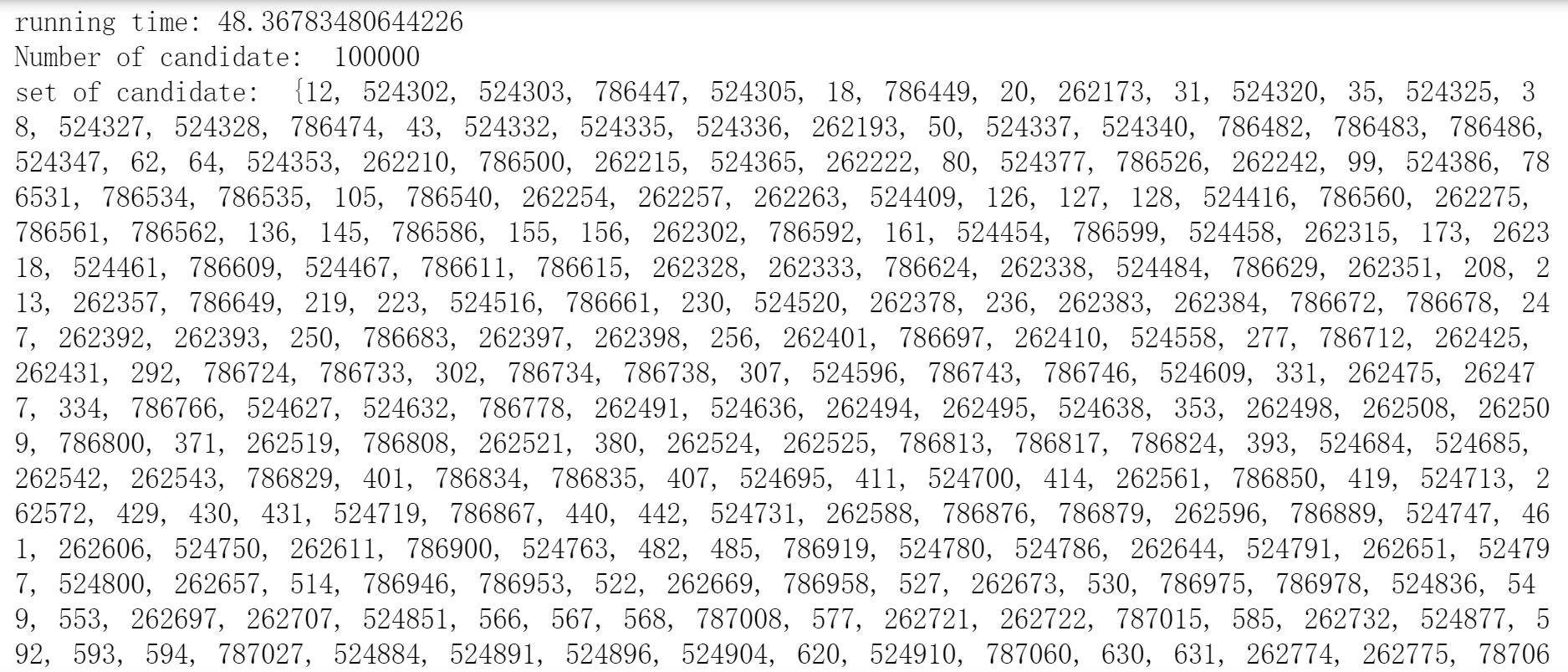
Simple1



Simple2



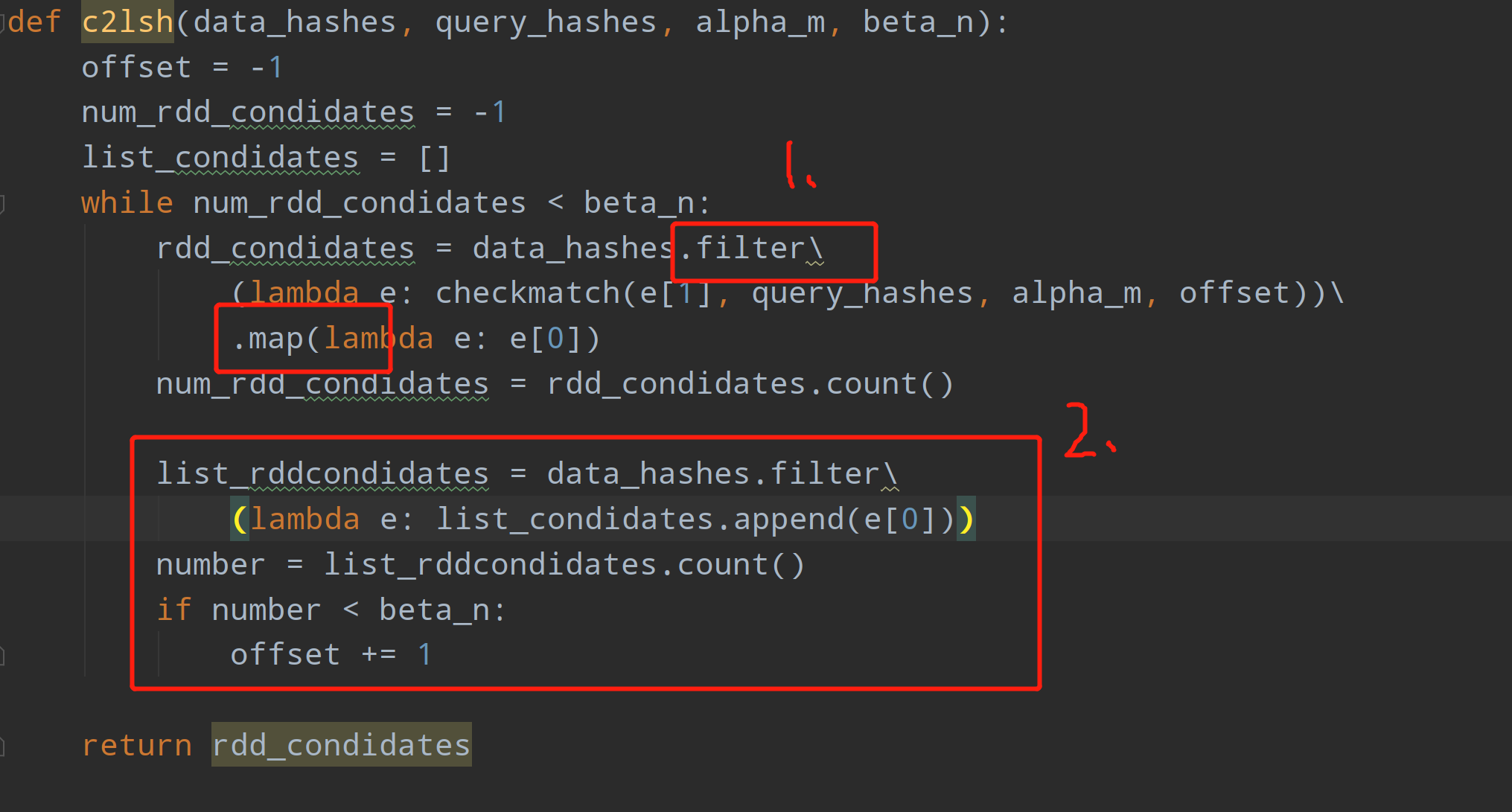
Simple 3

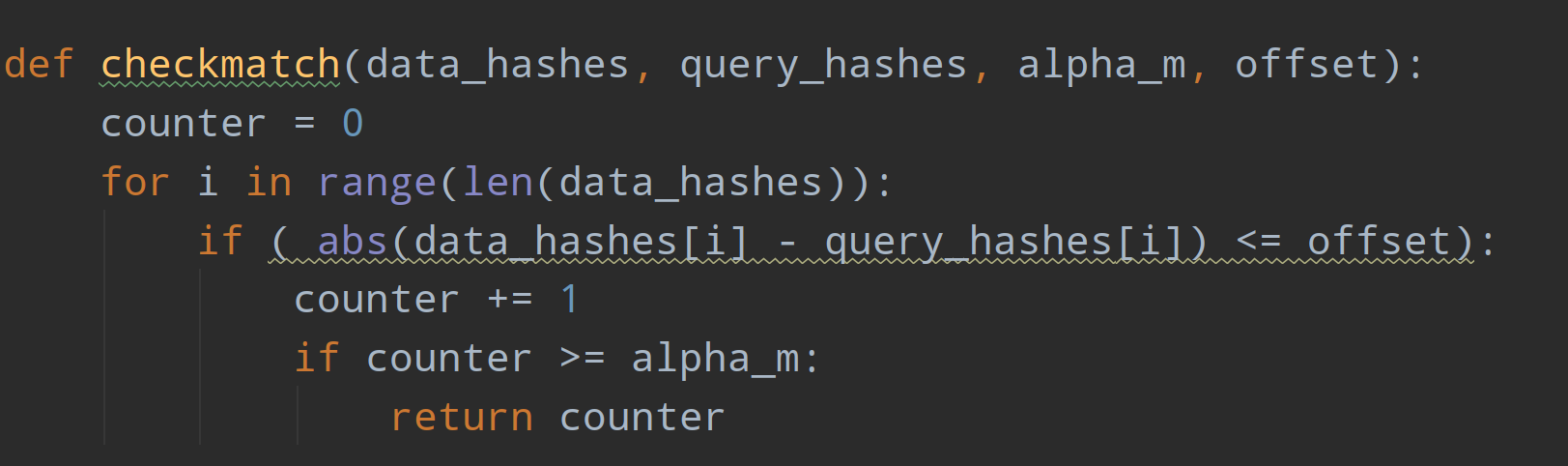


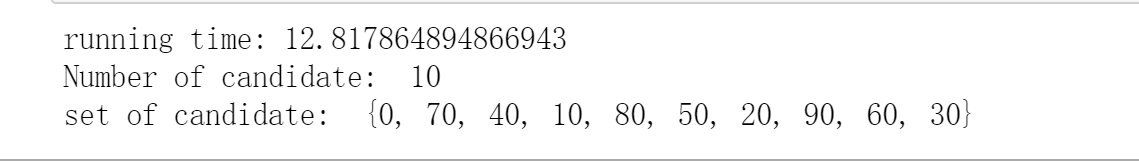
As the test data increases, the running time becomes longer, but the overall running speed is not very bad.

Part 4 : Improve the efficiency of my implementation

All tests and improvements are completed in the toy data set test. According to the pseudo-code given by lecture, I completed the initial code.

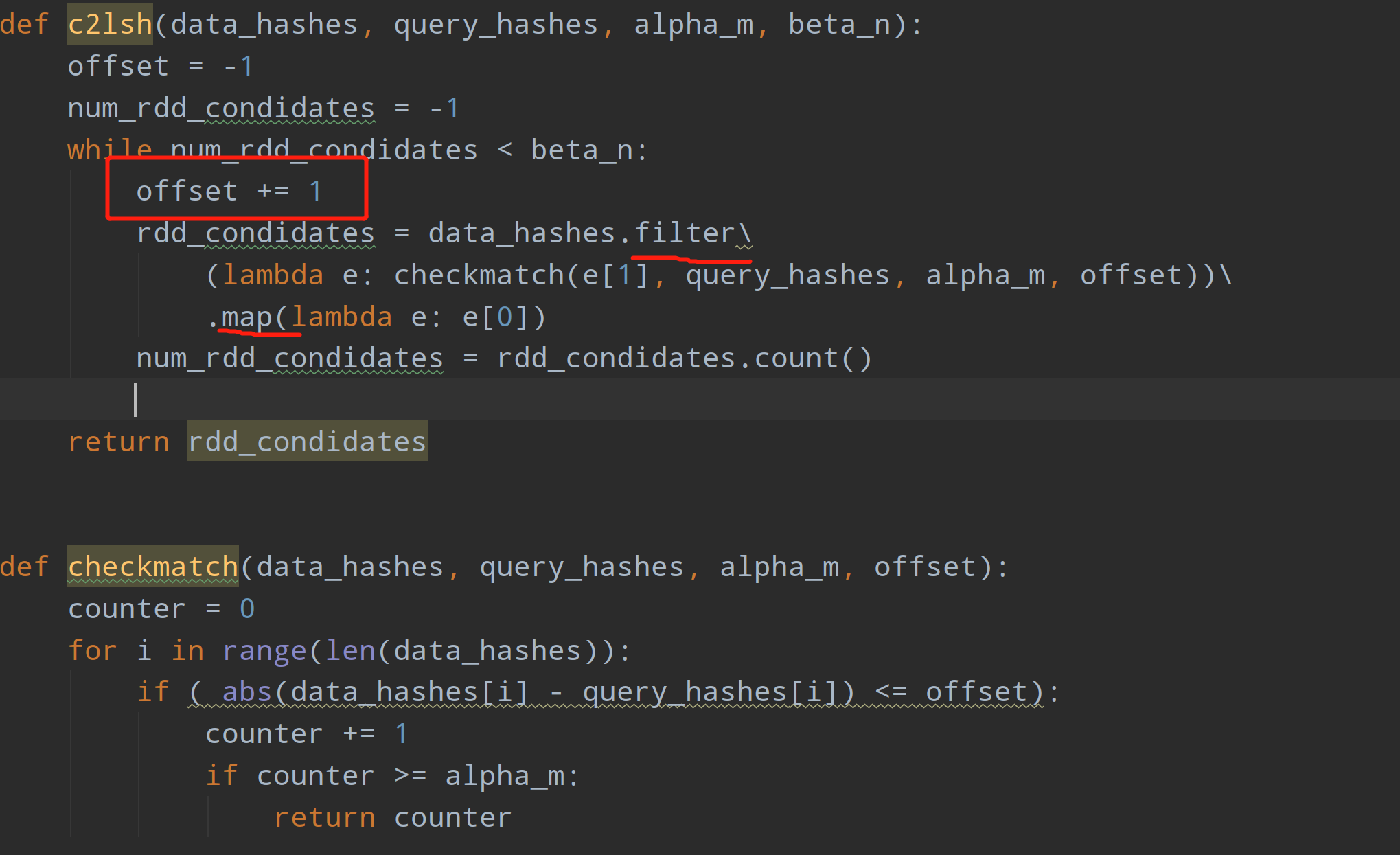


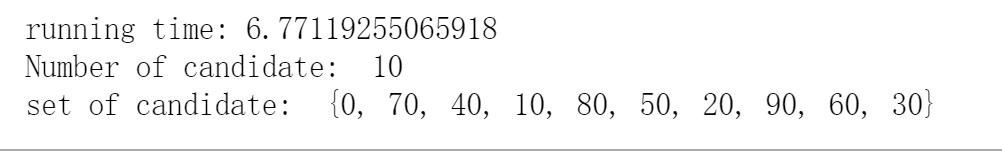




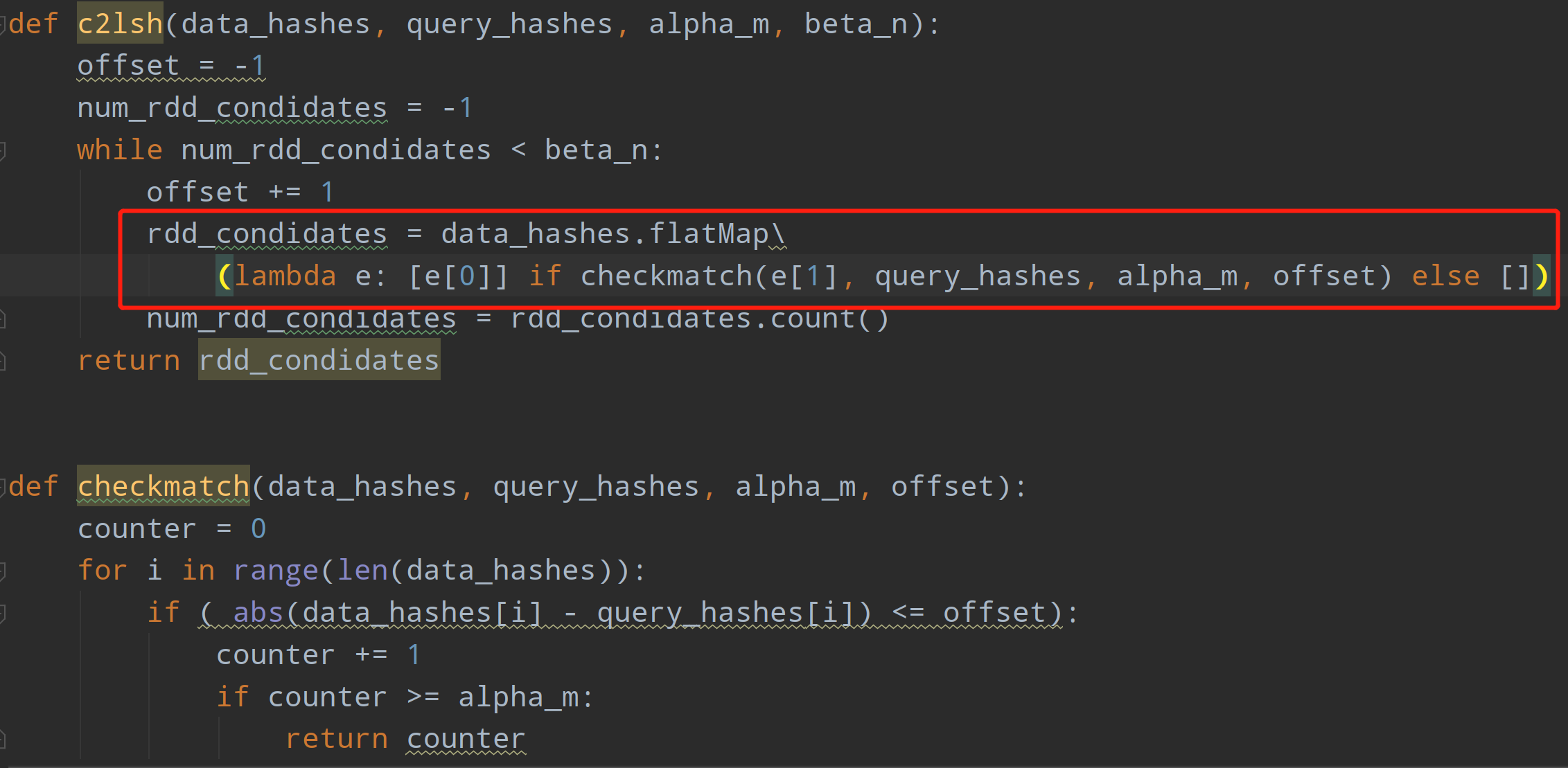
In this code, build filter and map method complete RDD translated, there have twice RDD translate. Then in order to add the item in to candidates and offset +=1, I did the RDD translated again. So in this code has three times RDD translated. The running time is 12.8 seconds.

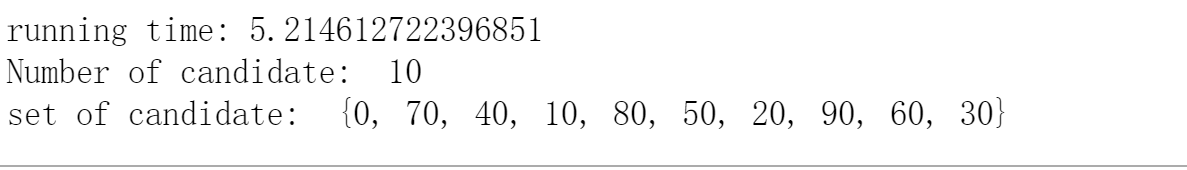
However I fund the number part 2(red number) is not necessary. So I did the first change code . I fount offset can follow while method, so I put the offset+=1 at the RDD translate front. Continue to use filter and map method translated RDD. There only has twice RDD translated. The result is better, running time is 6.7 seconds.





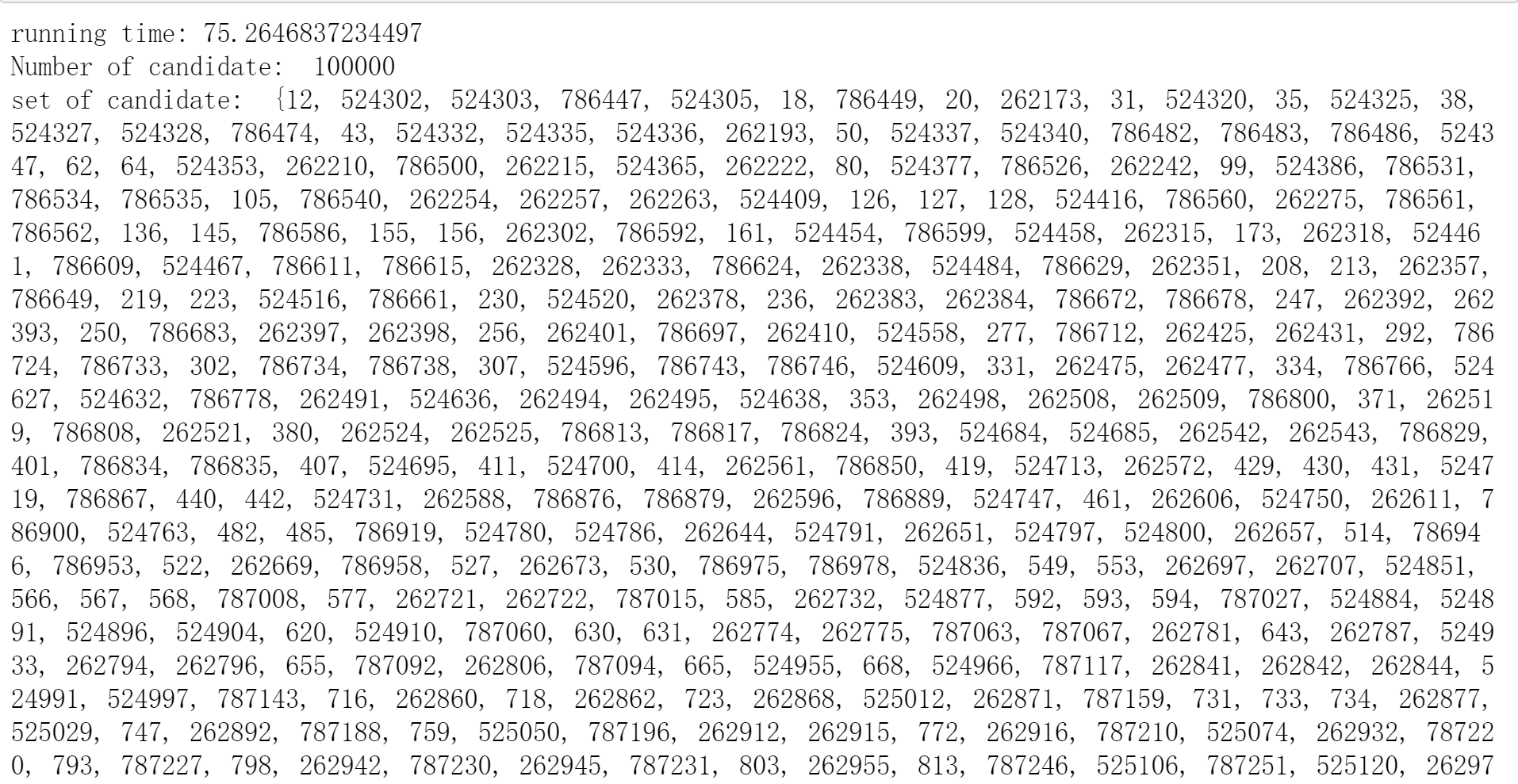
Secondly, using flatMap method replace the filter and map methods, because the flatMap method can achieve the same operation and get same result. And there is only one RDD translated here. The running speed will definitely increase.



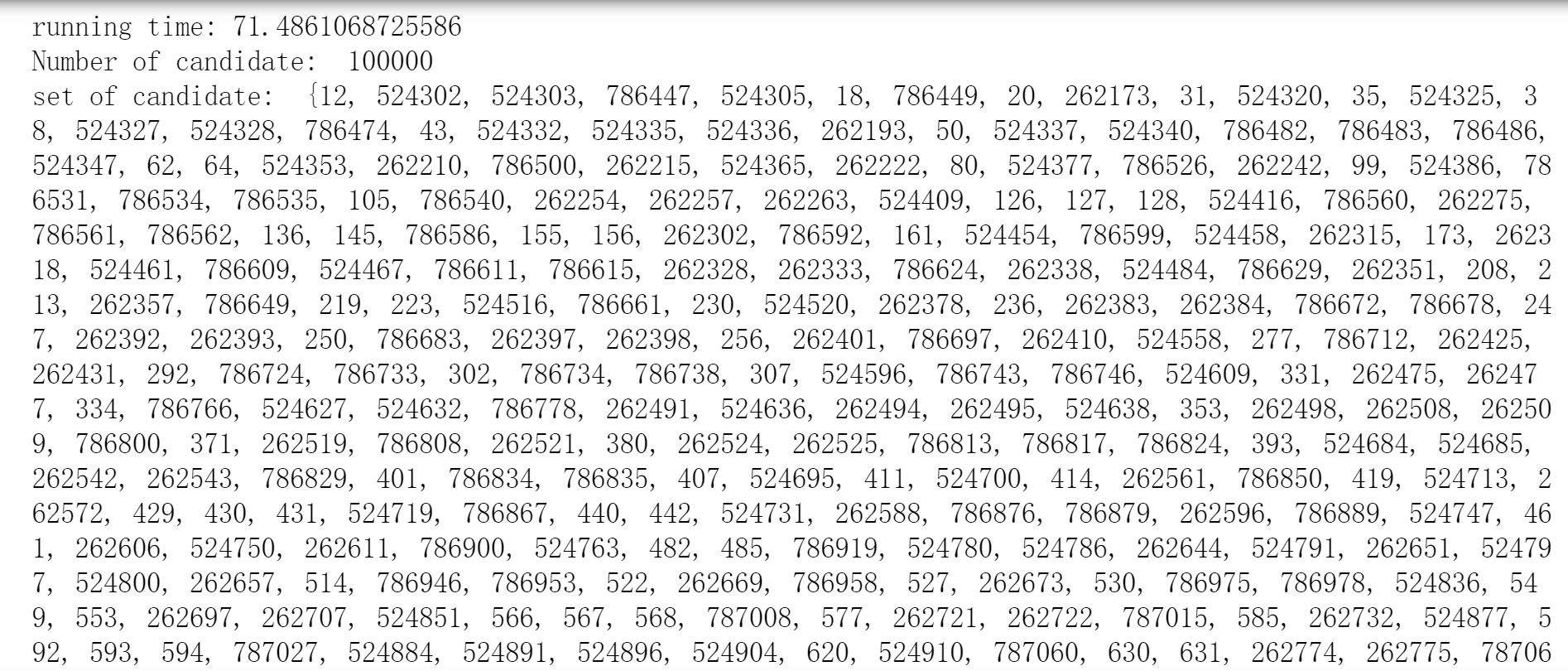


The result has indeed changed, but the effect is not particularly obvious. The reason may be that the test set is not large enough. Below I use my simple3 testcases, one million to test the effect of these three improvements.

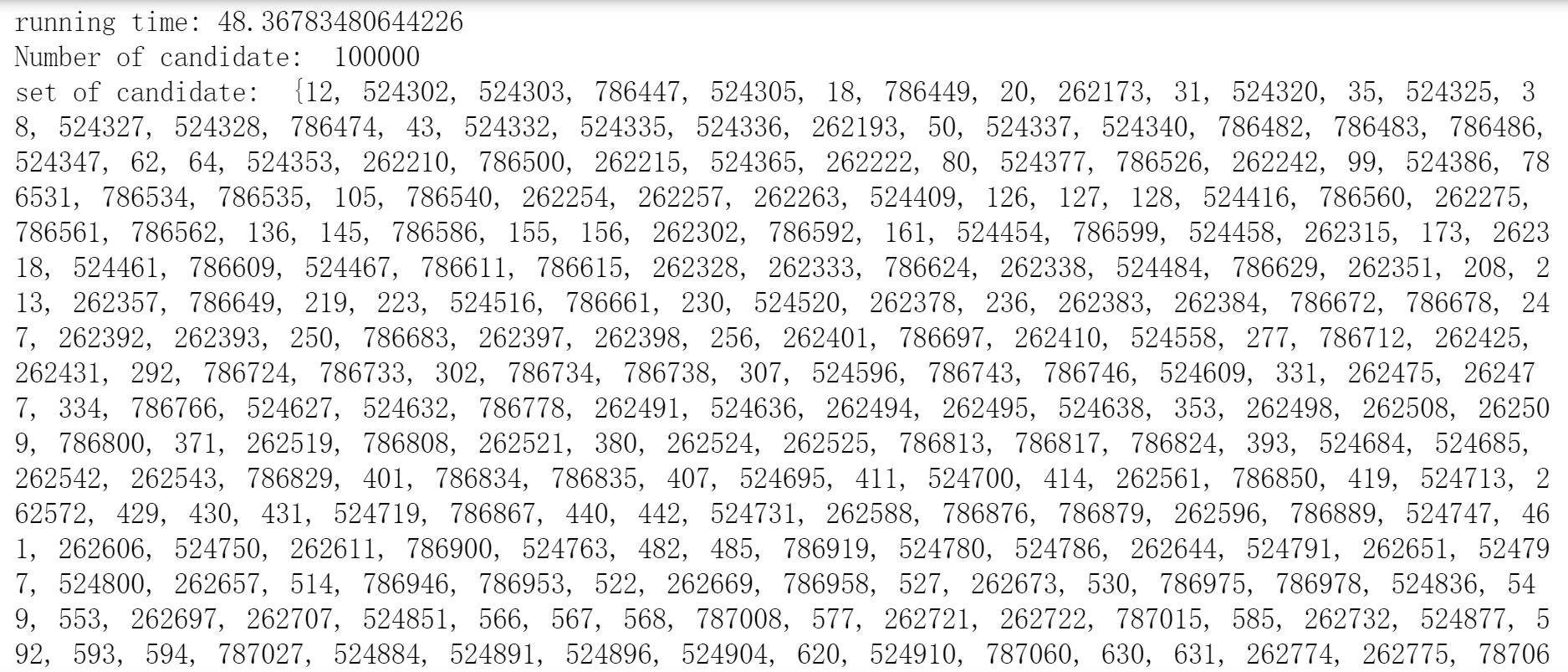
First



Second



Third



In the case of using a single core, the final code running time is shortened by 23s, the result is very good.