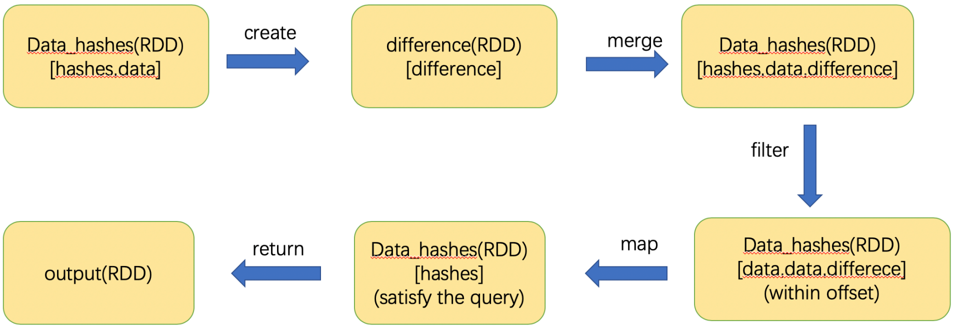
#COMP9313 project1 report
#ZANNING WANG
#zID: z5224151

Q1、Implementation details of your c2lsh(). Explain how your major transform function works.
Q2、Show the evaluation result of your implementation using your own test cases.
Q3、What did you do to improve the efficiency of your implementation?

Answer 1、



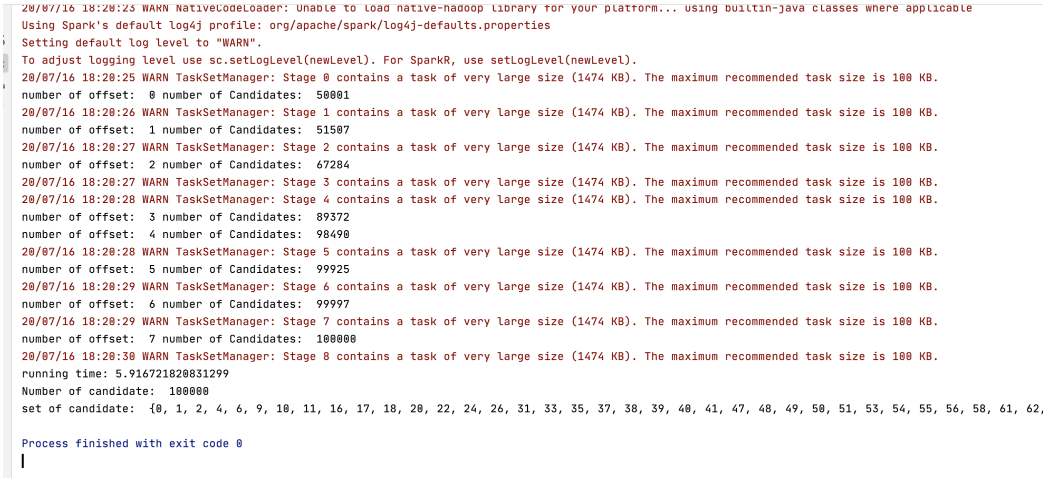
In order to find the rdd-id which match the query, under the c2lsh()function, I build another two functions called check_satisfy(), and check_diff(), the function check_satisfy() used to check the | data-query | < offset, if the data satisfy this equation, it will return true.; the fuction check_diff() used to create a new list which store the difference between data and query.

Use data_hashes.map to change the data_hashes from [id,data_hash] to [id, difference, data_hash

Use data_hashes.filter to filter the data similar to the query, after filtrate the data, the data_hashes(rdd) only contain the data satisfy the query.

Use data_hashes.map to map the data id and return to output

Answer 2、



```
20/07/16 18:20:23 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
20/07/16 18:20:25 WARN TaskSetManager: Stage 0 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  0 number of Candidates:  50001
20/07/16 18:20:26 WARN TaskSetManager: Stage 1 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  1 number of Candidates:  51507
20/07/16 18:20:27 WARN TaskSetManager: Stage 2 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  2 number of Candidates:  67284
20/07/16 18:20:27 WARN TaskSetManager: Stage 3 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
20/07/16 18:20:28 WARN TaskSetManager: Stage 4 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  3 number of Candidates:  89372
number of offset:  4 number of Candidates:  98490
20/07/16 18:20:28 WARN TaskSetManager: Stage 5 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  5 number of Candidates:  99925
20/07/16 18:20:29 WARN TaskSetManager: Stage 6 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  6 number of Candidates:  99997
20/07/16 18:20:29 WARN TaskSetManager: Stage 7 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
number of offset:  7 number of Candidates:  100000
20/07/16 18:20:30 WARN TaskSetManager: Stage 8 contains a task of very large size (1474 KB). The maximum recommended task size is 100 KB.
running time: 5.916721820831299
Number of candidate:  100000
set of candidate:  {0, 1, 2, 4, 6, 9, 10, 11, 16, 17, 18, 20, 22, 24, 26, 31, 33, 35, 37, 38, 39, 40, 41, 47, 48, 49, 50, 51, 53, 54, 55, 56, 58, 61, 62,

Process finished with exit code 0
```

In my own test case, I set alpha_m = 9, beta_n =100000, the test result show above.
Answer 3、

In order to improve the efficiency of my implementation, I create function check_diff(), to check the difference of data_hashes and query_hashes only one time and save it into the output, in this way, everytime we add offset to get more data satisfy the query, we do not need to compare the data and query again, instead of comparing the data and difference.

And I try to aggregate the data_hashes(rdd) with the same data to improve the efficiency, so that the same data in data_hashes(rdd) will be count at the same time, however, the in this progress, the function will create new rdd and merge it together, which will decrease the efficiency.