20171401, JAYITHA .C

DATABASE SYSTEMS

ASSIGNMENT 4
SORT MERGE JOIN AND HASH JOIN

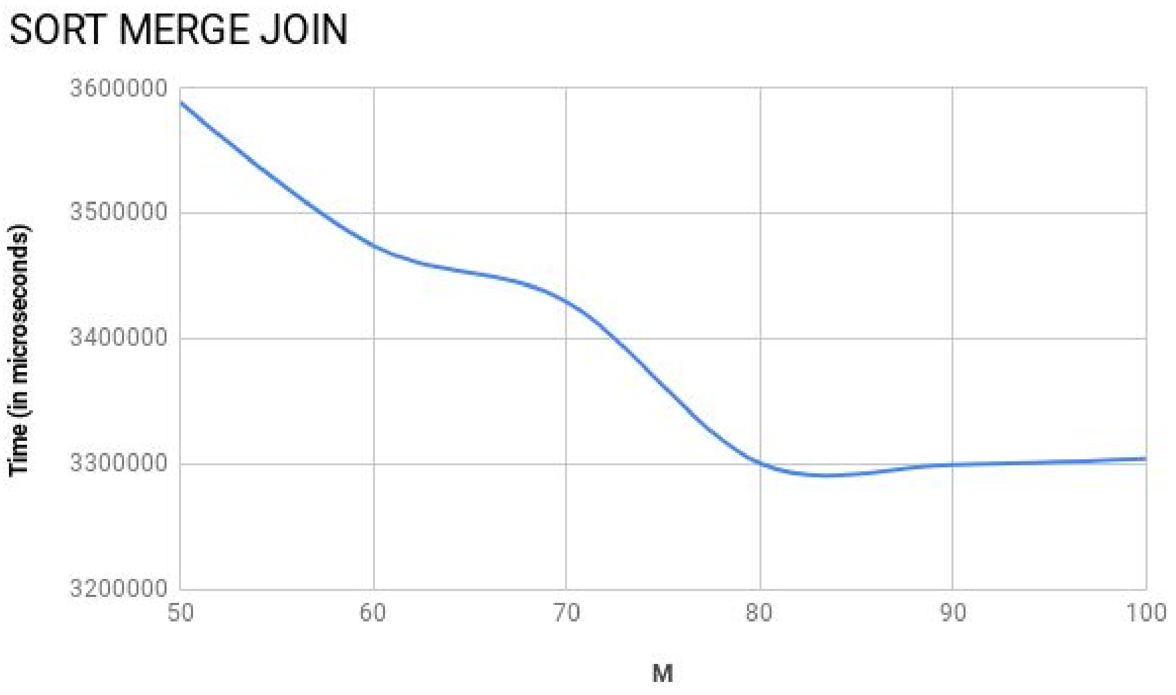
INPUT DATA FILES

Generated Input files R1.txt and S1.txt such that each has 50000 records (50 thousand records each) and performed sort merge and hash join procedures and measured the times taken.

SORT MERGE JOIN

Optimisations done: In the case of keys that are repeated a large number of times(one block size max) then a copy block has been implemented to handle this case

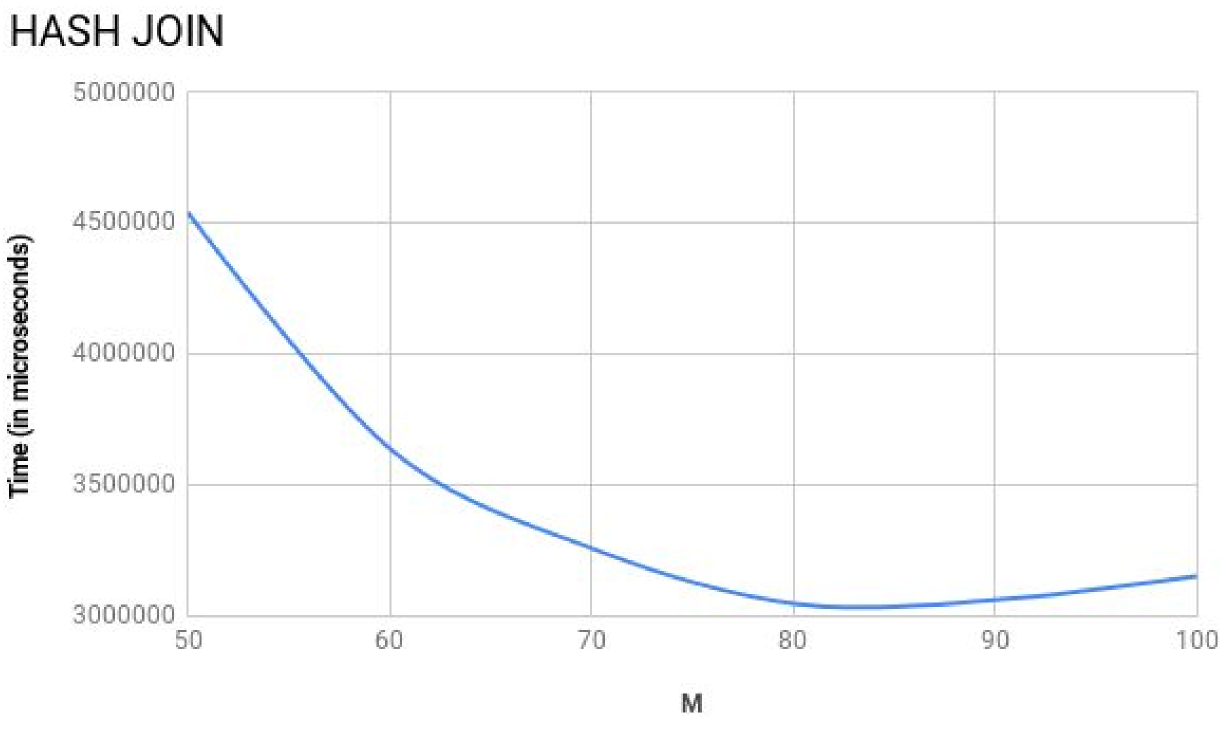
M	Time Taken (in microseconds)	Number of Rows
50	3588744	49846
60	3474430	49846
70	3429749	49846
80	3300959	49846
90	3299773	49846
100	3304468	49846



HASH JOIN

Optimisations done: Implemented grace hash join where the second hashing uses linear probing.

M	**Time Taken (in microseconds)**	**Number of Rows**
50	4541794	49846
60	3640397	49846
70	3258908	49846
80	3048217	49846
90	3061748	49846
100	3152521	49846



CONCLUSION

In conclusion, sort merge join seems to perform better for small values of M, but the bottle neck with sort merge join is when a given join key is repeated more than 100 times, then to match the join key we may need to store a temporary copy of all the matches (a version of which has been implemented in the code provided). Hash joins perform better for large values of M, but hashing is very susceptible to failure for some datasets because it is very dependent on the hashing functions used. For this assignment, I have implemented a modulus hash function for the first level hash and then a linear probing hashing mechanism for the second phase.