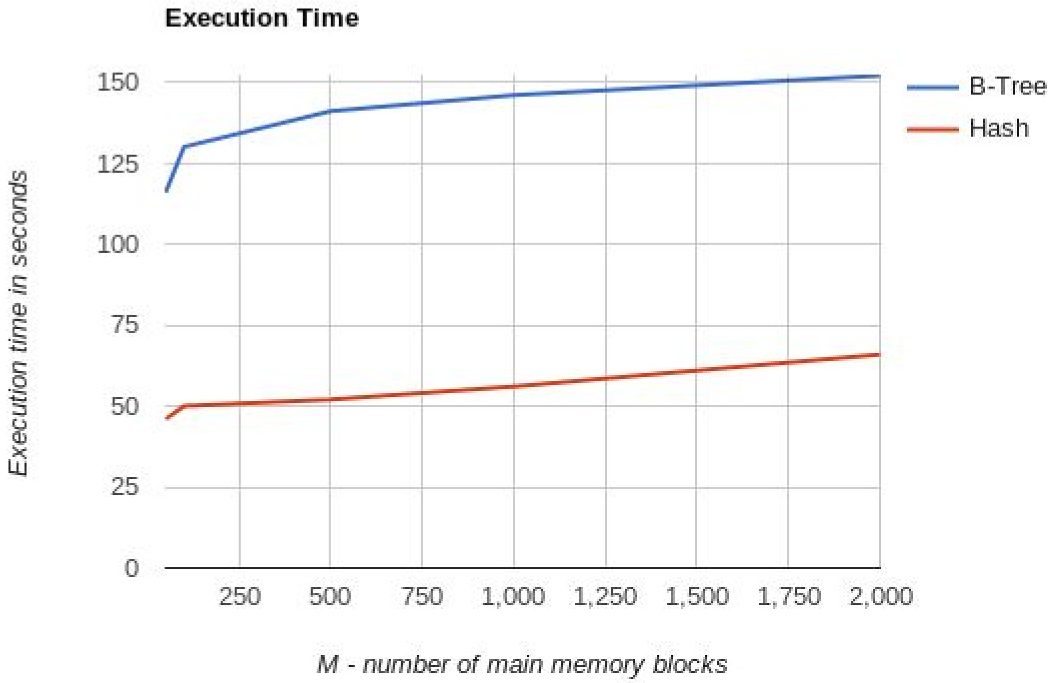
DB Duplicate Elimination Report

Chaitanya Patel - 201501071

- Block size : 32 KB
- Input file
  - number of rows : 10,000,000
  - duplicate : 40 %
  - number of cols : 4
  - each tuple size : 16 Bytes
  - B(R) = 4882

M (number of main memory blocks)	B Tree Index	Hash Index
50	real 1m56.115s user 1m54.840s sys 0m1.004s	real 0m46.588s user 0m44.092s sys 0m1.396s
100	real 2m10.235s user 2m9.541s sys 0m1.124s	real 0m49.781s user 0m47.412s sys 0m1.539s
500	real 2m21.293s user 2m19.268s sys 0m0.988s	real 0m52.055s user 0m51.936s sys 0m1.956s
1000	real 2m26.129s user 2m25.142s sys 0m1.973s	real 0m56.392s user 0m55.736s sys 0m1.536s
2000	real 2m32.542s user 2m30.552s sys 0m1.532s	real 1m8.560s user 1m6.076s sys 0m2.844s



- According to the table and graph, we can see that change in M doesn't affect much on the execution time of duplicate removal.
    - This can be explained by the fact that duplicate removal is one-pass algorithm. Also if we don't consider hash or btree, then it is also a tuple-at-a-time algorithm. Here, number of I/O operations doesn't change. Hence the execution time is same.
    - We notice slight increase in execution time with increase in M. With increasing M, number of buffers increases. Hence, time to find filled buffer slightly increases. Thus execution time slightly increases.