11 – Join Algorithms

1. Join operator's input and output is table (data)
2. Nested Loop Join
   1. Just using nested for-loop
      very inefficient : each records in R, should scan table S
   2. Block nested loop join
      less inefficient : each page in R, should scan table S
      if Buffer Pool is big enough to contain table S, it can be more efficient.
   3. Index nested loop join
      more efficient : no need to scan table S each time, search is more faster
3. Sort-Merge Join
   1. Two pointer algorithm
   2. Process
      1. Sort two tables
      2. Merge it with two pointer
         while R's cursor's key is smaller or equal to S's cursor's key, move S's cursor, if key are same, emit it
         if S's cursor's key is larger than R's cursor's key, reset the S's cursor, and move R's cursor.
   3. Roughly O(M+N), worst O(M*N)
   4. Useful when table is already sorted or output should be sorted on join key

4. Hash Join

   A. Hash and match

   B. Grace hash join

      i. Hash both tables
      if partition is large than one page, hash that partition again
      (recursive hashing)

      ii. For each partitions which are corresponding, do block nested loop join

   C. Cost is O(M*N)

      i. Partitioning phase : 2*(M+N), Probing phase : M+N

5. What is it?

   A. Bloom filters