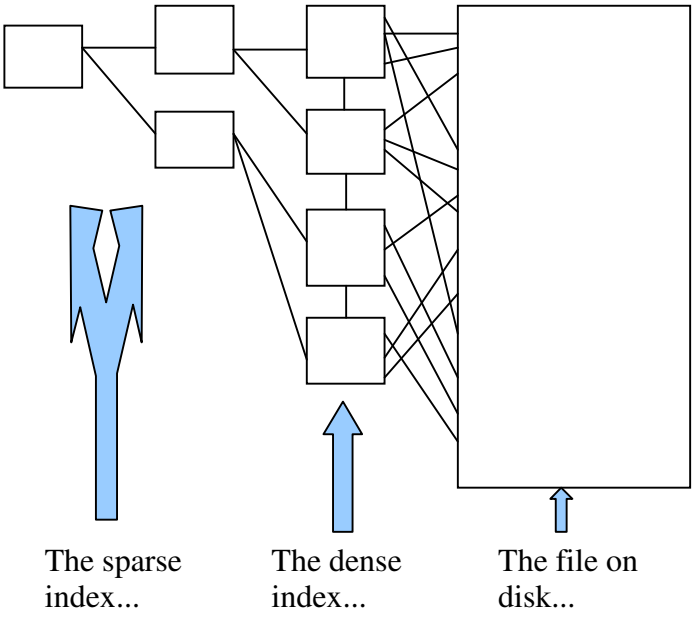
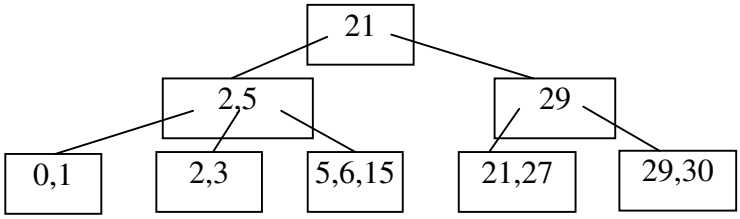
Example Solution (Updated from a preliminary version of Paea LePendu)
Database Processing CIS 451/551
November 26, 2006
Assignment 5

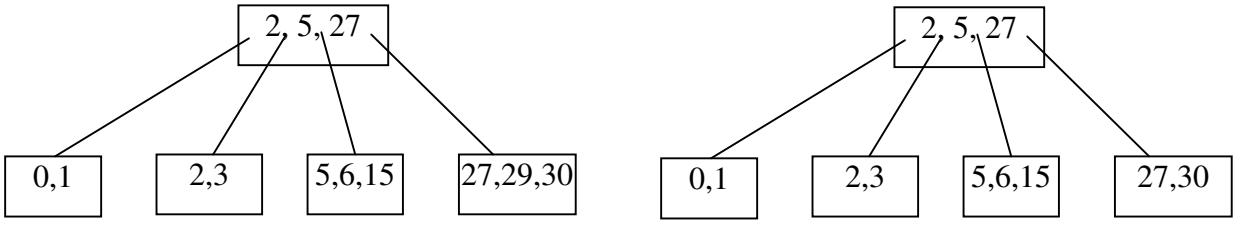
1. a) Primary indexes exist on files that are sequentially ordered on disk, so you can do a linear scan along with a sparse index to find records. Secondary indexes are randomly ordered on disk, so such a scan is not possible.
   b) Below is a rough depiction of a multi-level index. If you rotate it 90-degrees, it is just like the concept of a B+ tree.



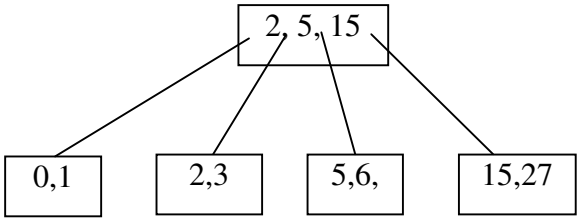
2. Insert: 30,0,29,1,27,2,21,3,15,5,6



a) Delete 21, 29:

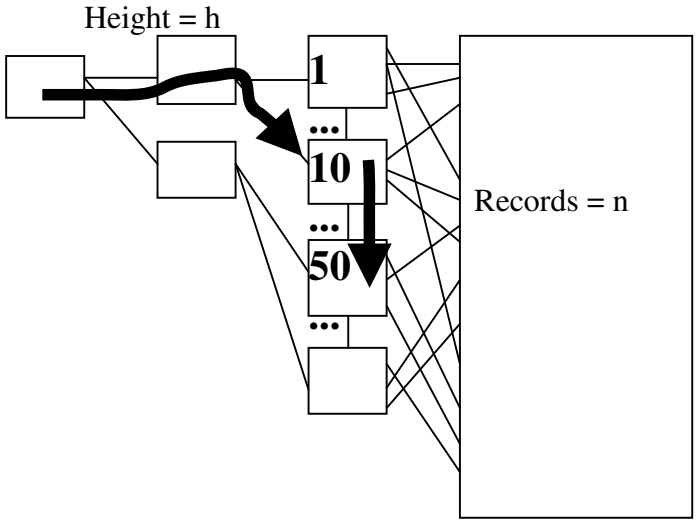


b) Delete 30:



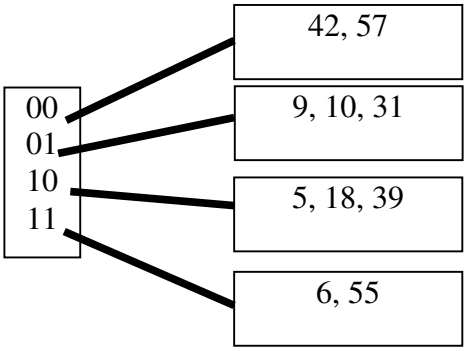
3. a) The idea behind O(n + h) is that we only have to search through the B+ Tree index once for a total of h internal node retrievals, then we just scan through the leaf nodes one at a time left-to-right n times. For an illustration, the image below is the idea of the search path for $10 < X < 50$.

   Please be careful not to confuse a linear scan of the index's leaf nodes versus a linear scan of a sequential file! They are similar concepts, but the aim is different.



   b) Hash index doesn't require the records are sequential. We have to access each element individual for the queries in the question. The worst case the running time is O(nk) where k is the size of the bucket.

4. We only need to use 2-bits.



5. Without giving you a definitive answer for this problem, if you compare solutions to problem #4 above with the similar problem from last year using *h(x)= x mod 15*, the evidence would seem to support the proposition that a prime divisor is better (because we use less bits with *h(x) = x mod 7*). But evidence is not proof. There are exceptions, which depends on the distribution of records.