Chapter 1

Refinement of Indexes and B+ Trees

Let us consider issues that exist in any index structure (not just B+ trees):

- Query Support: what class of queries does the index allow?
- Choice of Search Key: affects queries for which we can use an index.
- Data Entry Storage: affects performance of the index
- Variable-length key tricks: affects performance of the index.
- Cost model for index vs. heap vs. sorted file

1.1 Query Support

Usuaully indexes are used for basic select queries, including equality selection (operation is '=') and range selection (operation is one of ¡. ¿, ¿=, ¡=, BETWEEN). B+ trees provide queries for both of these on a single dimension. There are other, more exotic, selections. There are queries for 2-d boxes or circles, for n-dimensional indexes a la R-trees and KD-trees. These are hard to perform well when we get to lots of dimensions ("Curse of Dimensionality" - will be reviewed at a later time). There are also near-neighbor queries ("10 restaurants closest to the Empire State Building."). There are also regular expression matches, genome string matches (matching protein pairs), and more. The rest of this chapter will focus on taditional 1-d range search with equality as a special case.

1.2 Search Key and Ordering

We can index on any ordered subset of columns