Hash-Based Indexing

- Records in a file are grouped into **buckets**
- Each bucket consists of a **primary page** and zero or more **overflow pages**
- A **hash function** h takes a search-key value k and returns the address of a bucket
- To find records matching a search key value k, calculate h(k), then look through bucketed pages sequentially to find matching data entries

Tree-Based Indexing

- Search key values are organized in a **tree**
- The highest level is the **root**
- The lowest level (the **leaf level**) contains data entries
- Each node in the tree is a page on disk
  - Retrieving nodes involves disk I/O
  - And therefore the number of disk reads in a search is equal to the length of the path from root to leaf
- A **B+ tree** is an index structure that ensures all paths from root to leaf are the same length

Comparing File Organizations

- We are interested in the *total cost* of accessing and modifying data with a given file organization scheme
- Specifically, what is the cost of:
  - **Scan** (fetch all records in a file)
  - **Search with equality selection** (fetch records that match an equality condition)
  - **Search with range selection** (fetch records that match a range condition)
  - **Insert** (insert a new record into a file)
  - **Delete** (delete a record from a file)

Cost Model

- To estimate cost, we need a model of total execution time
- Our model is a simplified one:
  - B is the number of pages (assuming 100% capacity)
  - R is the number of records per page (100% capacity)
  - D is the average time to read/write a page from/to disk
  - C is average time to process a record
- Consider the *average case*
- This is good enough to indicate trends

Heap Files

- Heap file = randomly ordered records
- Costs:
  - Scan: B(D+RC)
  - Search with equality selection: 0.5B(D+RC)
    - (if equality field is a key; same as scan if not)
  - Search with range selection: B(D+RC)
  - Insert: 2D+C
  - Delete: search cost + C+D

Sorted Files

- Records stored directly, sorted on one or more fields
- Costs:
  - Scan: B(D + RC)
  - Search with equality selection: $D \log_2 B + C \log_2 R$
    - (assuming selection field is the sort field)
  - Search with range selection: $D \log_2 B + C \log_2 R$
  - Insert: search + B(D + RC)
  - Delete: search + B(D + RC)

