Auxiliary structures that speed up operations that are not supported efficiently by the basic file organization.

A Simple Index Example

Entries in an Index

Organization of Index Entries

From BST to BBST to B
B-tree (B+-tree) Example

B-tree Properties
- Each node occupies one block
- Order n
  - n keys, n+1 pointers
- Nodes (except root) must be at least half full
  - Internal node: \[\lceil \frac{n+1}{2} \rceil \] pointers
  - Leaf node: \[\lceil \frac{n+1}{2} \rceil \] pointers
- All leaf nodes are on the same level

B-tree Operations
- Search
- Insert
- Delete

B-tree Insert
- Find the appropriate leaf
- Insert into the leaf
  - there’s room \(\rightarrow\) we’re done
  - no room
    - split leaf node into two
    - insert a new <key,pointer> pair into leaf’s parent node
- Recursively apply previous step if necessary
  - A split of current ROOT leads to a new ROOT

B-tree Insert Examples
- (a) simple case
  - space available in leaf
- (b) leaf overflow
- (c) non-leaf overflow
- (d) new root

(a) Insert key = 32
n=3
(b) Insert key = 7

(c) Insert key = 160

(d) New root, insert 45

B-tree Delete

- Find the appropriate leaf
- Delete from the leaf
  - still at least half full → we're done
  - below half full – coalescing
    - borrow a <key,pointer> from one sibling node, or
    - merge with a sibling node, and delete from a parent node
- Recursively apply previous step if necessary

B-tree Delete in Practice

- Coalescing is usually not implemented because it's too hard and not worth it

Static Hash Index

- Record
- Hash function
- Bucket directory
- Bucket blocks
- Overflow blocks
- Hash Index
Hash Function

- A commonly used hash function: $K \mod B$
  - $K$ is the key value
  - $B$ is the number of buckets

Static Hash Index Example ...

- 4 buckets
- Hash function: $key \mod 4$
- 2 index entries per bucket block

... Static Hash Index Example

- Insert the records with the following keys: 4, 3, 7, 17, 22, 10, 25, 33

Dynamic Hashing

- Problem of static hashing??
- Dynamic hashing
  - Extendable Hash Index

Extendable Hash Index ...

- Maximum $2^M$ buckets
  - $M$ is maximum depth of index
- Multiple buckets can share the same block
- Inserting a new entry to a block that is already full would cause the block to split

... Extendable Hash Index

- Each block has a local depth $L$, which means that the hash values of the records in the block has the same rightmost $L$ bit
- The bucket directory keeps a global depth $d$, which is the highest local depth
Extendable Hash Index

Example

- \( M = 4 \) (i.e. could have at most 16 buckets)
- Hash function: \( \text{key} \mod 2^4 \)
- 2 index entries per block

Extendable Hashing (I)

Bucket directory

Bucket blocks

insert 8 (i.e. 1000)
Insert 11 (i.e. 1011)
insert 4 (i.e. 0100)

Extendable Hashing (II)

Bucket directory

Bucket blocks

insert 14 (i.e. 1110)

Extendable Hashing (III)

Bucket directory

Bucket blocks

Readings

- Textbook Chapter 21.1 – 21.4