Midterm Review – Part I

(Disk, Buffer and Index)

CS186 Spring 03

Content

- Quick Review
- Questions from You
- Disk Example
- Buffer Example
- B+ Tree Example (1)(2)

Quick Review

- Disk
  - Components of disk
  - Disk access time
    - seek + rotation delay + transfer
  - Disk space management
    - Heap file (list / page directory)
    - Record format (fixed length / variable length)
    - Page format - fixed / variable (slot directory)

Quick Review – cont.

- Buffer Management
  - Buffer pool
    - pin / unpin, dirty, etc.
  - Buffer replacement policy (LRU, MRU, Clock)
    - How they works?
    - Advantage?
    - Problem?
      - e.g. sequential flooding in LRU

Quick Review – cont.

- File Storage
  - Heap / Sorted / Clustered file
  - I/O cost of operation (see next page)
  - Basics about indexes
    - Why use? Speed up selection on keys.
    - Classification
    - Clustered vs. Unclustered
      - Related to cost!
      - How indexes are organized?
      - Pros and cons?

Quick Revie

Empirical study show that pages are about 67% occupancy. → The number of physical data pages is about 1.5B

	Heap File	Sorted Fil	
Scan all records	BD	BD	1.5 BD
Equality Search	0.5 BD	$(\log_2 B) * D$	$(\log_F 1.5B) * D$
Range Search	BD	$[(\log_2 B) +$ #match pg$]*D$	#match pg)
Insert	2D	$((\log_2 B)+B)*D$	$((\log_F 1.5B)+1) * D$
Delete	0.5BD + D	$((\log_2 B)+B)*D$ *(because R,W)*	$((\log_F 1.5B)+1) * D$

Ignore the case when leaf is full, so only one write here.

2 * 0.5BD (read+write) for moving the later parts of the file

B: The number of data pac
D: (Average) time to read

Quick Review – cont.

- Tree Structured Index
  - ISAM
    - How the index is organized?
    - Know about how to insert, delete …
  - B+ Tree
    - Features (pros and cons)
    - How the index is organized?
    - Familiar with bulk load, insert, delete …
    - Simple computation using fanout, order, height …

Questions From You

- For variable-length records page format (lecture 3), do we pack on removal? Doesn't seem like it would be an easy or cheap operation... But how else can we reclaim free space?
  - Do not pack each time. Pack the space only when not enough free space.
- Dense vs. sparse index?
  - Not in the coverage of this semester

Disk Example

- (Fall 1999 Midterm)

  Consider a disk with a sector size of 512 bytes, 100 sectors per track. Given a rotational speed of 7200 revolutions per second, what is the maximum rotational delay to the start of a sector? Assuming that one track of data can be transferred per revolution, what is the transfer rate?

- Maximum rotational delay:1/7200 sec
  - Average rotational delay: 0.5*(1/7200)
- Transfer rate: 512*100*7200 bytes/sec.

Buffer Example

- (Fall 1999 Midterm)

  Assume that in the CUSTOMERS table, records are much larger than shown in the table on the page – in fact, each record fits in exactly one page, so the CUSTOMERS table is 5 pages long. And also assume that there is room for only two buffers – that is, there are two buffer frames – in memory, and that these buffers start out empty.

  - Is LRU or MRU a better strategy if we are reading the CUSTOMERS records in the order: Cid = 1, 1, 3, 4, 3, 1, 2. Explain your answer.

Buffer Example – cont.

- 1, 1, 3, 4, 3, 1, 2

LRU

1	1	1	4	4	1	1
		3	3	3	3	2

MRU

1	1	1	1	1	1	2
		3	4	3	3	3

B+ Tree Example (1)

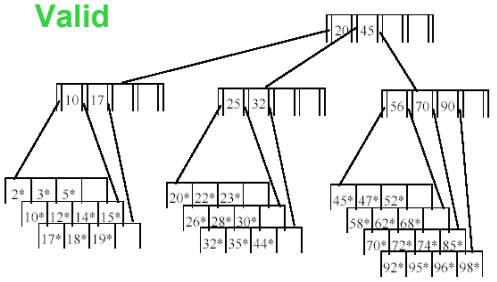
- (Fall 2002 Final)

  For each of the following B+ Trees, decide whether it is a valid B+ Tree (i.e., one that could exist after numerous inserts and deletes) or if it is invalid. Circle your choice, and if it is invalid, describe in one sentence the single main reason why.

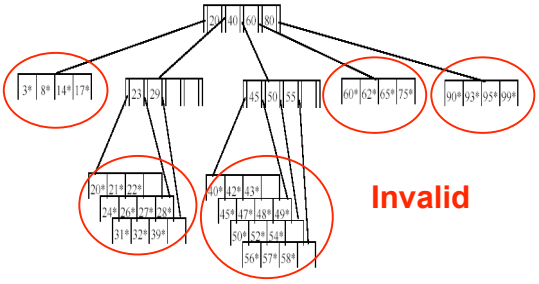
  - The trees follow all rules in the book including merging on delete.
  - All of the trees are of order d=2.

B+ Tree Example (1) – cont.

Valid

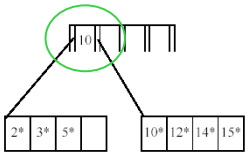


B+ Tree Example (1) – cont.



Invalid

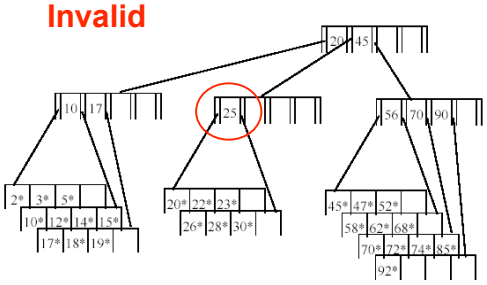
B+ Tree Example (1) – cont.



Valid

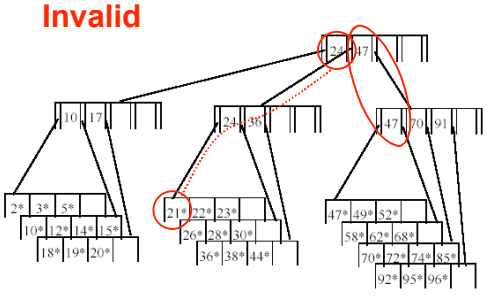
B+ Tree Example (1) – cont.

Invalid



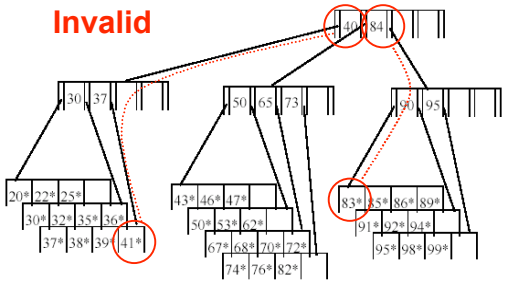
B+ Tree Example (1) – cont.

Invalid



B+ Tree Example (1) – cont.

Invalid



B+ Tree Example (2)

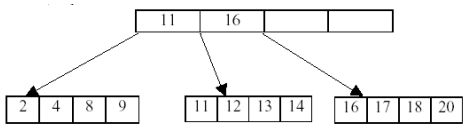
- (Spring 2002 Midterm)

  Consider a B+ tree containing the elements 2, 4, 8, 9, 11, 12, 13, 14, 16, 17, 18, 20. The tree has order d = 2. This means that internal nodes have at least 2 entries and 3 pointers and at most 4 entries and 5 pointers; leaf nodes have at least 2 entries and at most 4 entries.

B+ Tree Example (2) – cont.

- bulk-load

  (2, 4, 8, 9, 11, 12, 13, 14, 16, 17, 18, 20)



B+ Tree Example (2) – cont.

- Based on your original tree from above, what is the minimum number of deletions required to change the tree into one single leaf node?