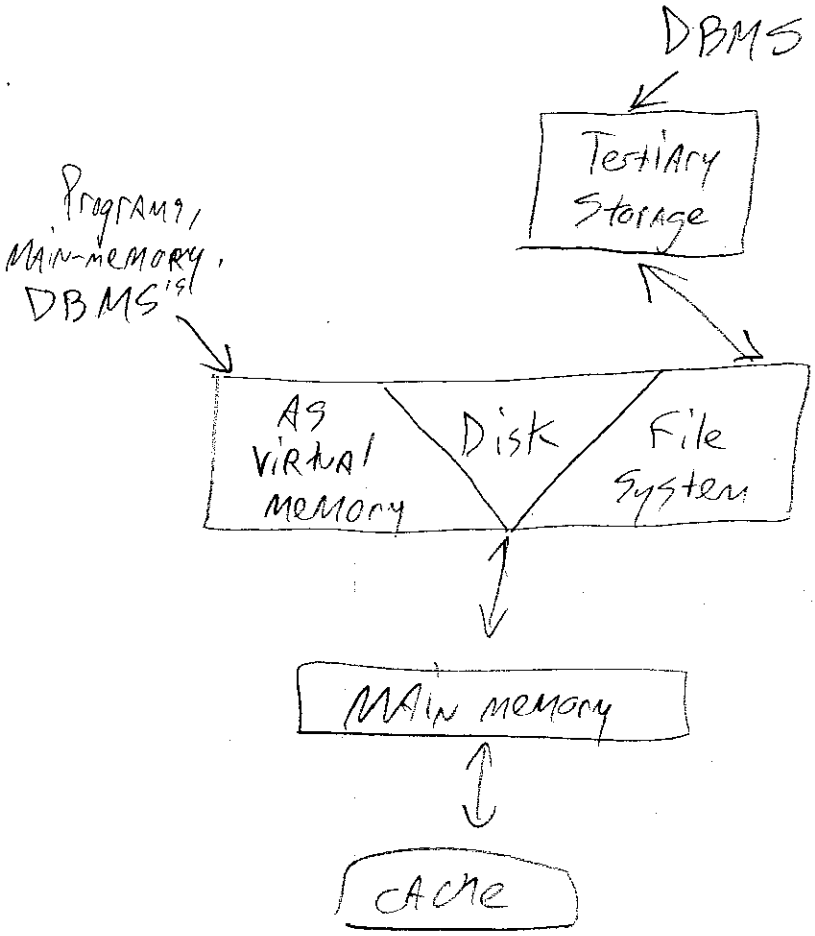
Sample Midterm 1

① Diagram for the memory heirarchy



Derreck McCray
Sonyajit Jena
Marvin Mante

#2

1,000,000 blocks

4k per block

100 MB of main Memory

A = Must calculate the total number of blocks that fits in main memory at one time:

$$\frac{100\text{ mb Main Memory}}{4\text{k per block}} = \frac{100 \times 2^{20}}{2^{12}}$$

$= 100 \times 2^8$ blocks in main memory at one time.

B = For reference, calculate the total number of times 100×2^8 blocks must be in main memory for 1,000,000 blocks to be sorted in the first phase

$$\frac{1{,}000{,}000}{100 \times 2^8} = \frac{10^6}{2^8} \approx 39$$

Phase 1: Each block is read and written, meaning that since there are 1,000,000 blocks, there are a total of 1,000,000 x 2 Disk I/O's.

Phase 2: Each block is once again read and written, also stating that there will be 2,000,000 more I/O's for the second phase.

All-in-all = 2,000,000 + 2,000,000 I/O's = 4,000,000 Disk I/O's

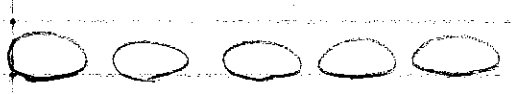
* Phase 2 may have had more I/O if B > A.

3) elevator scheduling

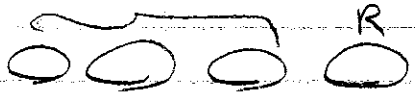
disc head receives the disk address from the first request and moves toward that disk block. Since new requests can be made as the disc head is moving, it has to check track by track whether a new request has been made there and process them. It will only check for new requests in the same direction until there are no more in that direction. Only then will it start checking for requests in the opposite direction.

Vasuki Kalluri
Phuong Nguyen
Sithu Aung
Kae Uchida

(4) RAID level 5 Vs. RAID level 4



n=5 0 1 2 3 4 (Remainder)



Disk_1_blk_1:

For Raid level 5 there is no seperate redundant disk. Every disk is treated as a redundant disk for some of it's cylinders. Which cylinders are redundant is based on the modulo division by the n where n = no of disks. So all the cylinder numbers which give a remainder 0, when divided by n, are considered redundant for disk 0. All the cylinder numbers which give a remainder 1, when divided by n are considered redundant for disk 1 and so on....

In Raid level 4 a seperate disk is assigned to be the redundant disk and the remaining disks are treated as data disks.

If any of the disks crashes (including the parity disk) they can be calculated by using the remaining disks.

For example:-

Disk1: 1000
Disk2: 1111
Disk3: ???? 0101
Disk4: 0010

Disk 3 : 0101

Raid level 4 and 5 can recover from single disk Crashes.

5

Recoder header

① pointers to the schema of the recoder
② the total length of the recoder
③ the timestamp of the recoder

Block header

① logical address of the block
② offset table to each recoder.
③ the used size of the block

#6 AUTOMATIC SWIZZLING TRANSLATES ALL DATABASE ADDRESSES INTO MEMORY ADDRESSES WHEN A BLOCK IS BROUGHT INTO THE MEMORY. SWIZZLING ON DEMAND TRANSLATES ONLY THE ADDRESSES THAT ARE DEREFERENCED, ONE AT A TIME.

(7) 50 records for Table T fit into a 4k block.

Index entries take 16 bytes each, but cannot use the last 64 bytes in a block

If T takes 500 million blocks to store

how many blocks will the index take:

(a) for a dense index ?

how many index entries do we need?

500 million blocks * 50 records/block = 25,000,000,000 records

how many block of indexes? $\frac{25,000,000,000 \text{ indexes}}{252 \text{ index per block}}$ =

99,206,350

(b) for a sparse index

how many indexes in a block?

block is 4096 bytes

− 64 for the amount not used

4032

÷ 16 for the size of index

252 index per block

a sparse index requires one entry per block:

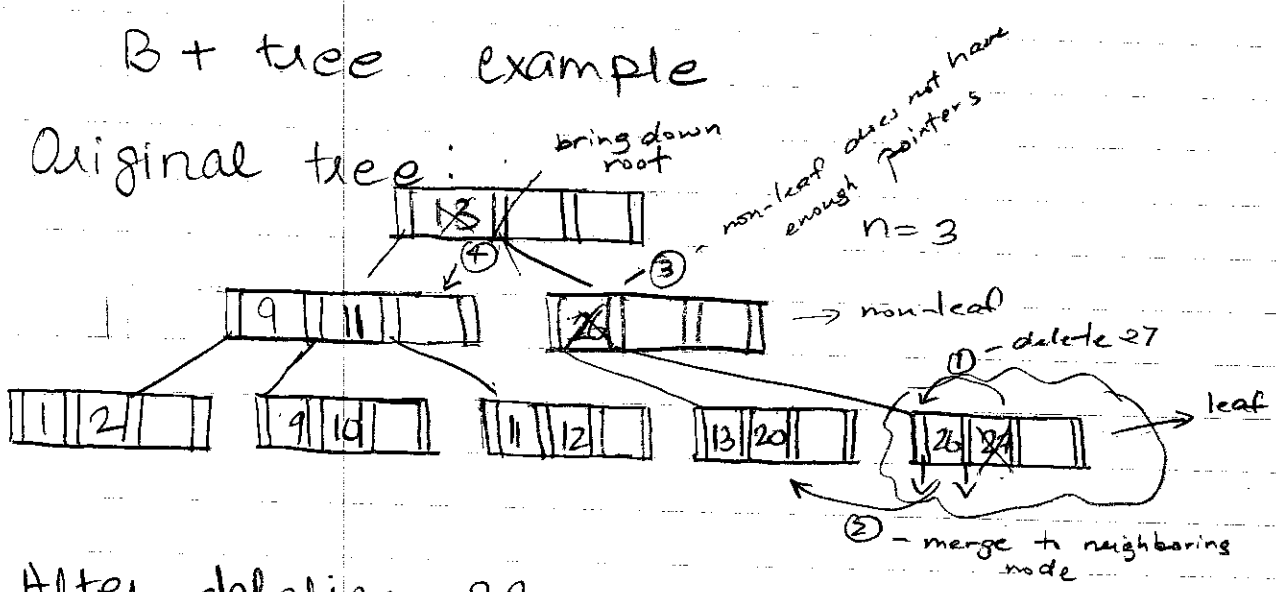
$\frac{500,000,000 \text{ blocks requiring indexes}}{252 \text{ index per block}}$

= 1,984,127

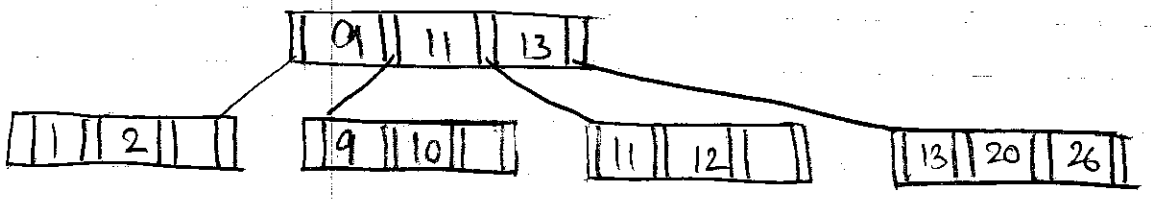
RWC

#8 B+ tree example

Original tree:



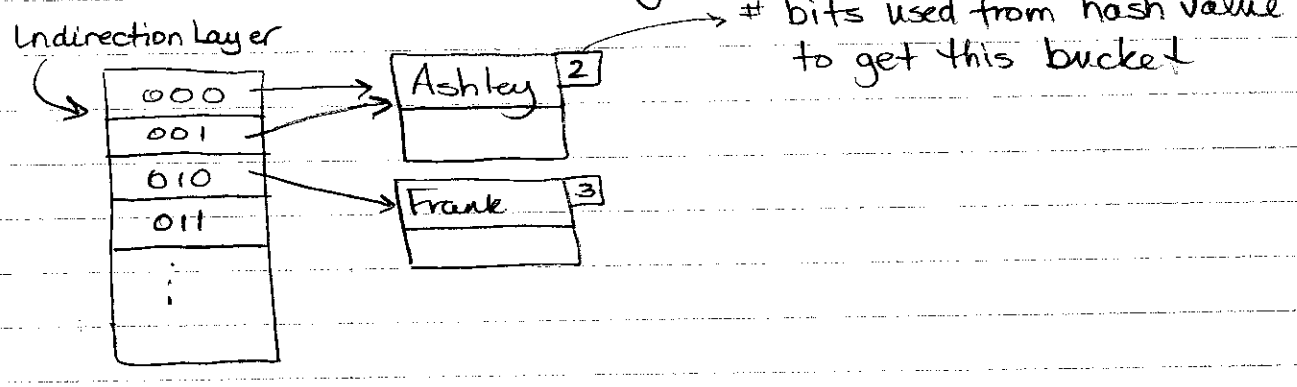
After deleting 27



3. The elevator scheduling algorithm is effective when there are a large number of Disk Requests arriving continuously. The disk head will traverse the Disk in one direction servicing any Request on each cylinder it passes. Once all requests, in its current direction, have been Resolved, the head turns around and completes the same process in the opposite direction

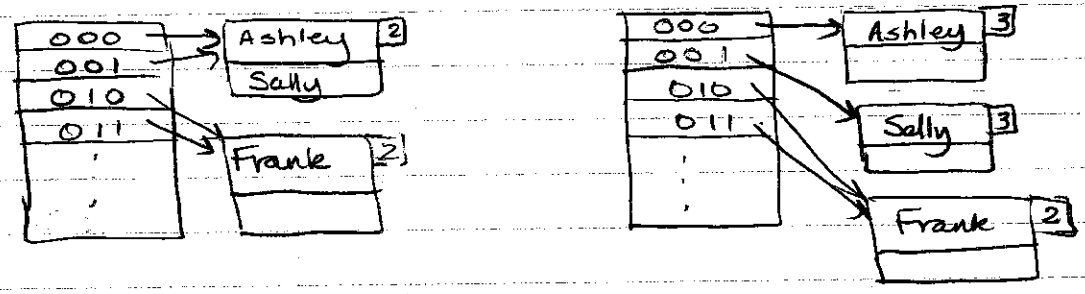
9. B trees are useful for queries in which a relatively narrow search range is requested from a large table
Or if one has a range query which can be answered using the index alone

(10) # Extensible Hashing



- Each bucket has a level of indirection associated with it.
- Indirection layer consists of bit strings + pointers that point to data buckets, more than one pointer can point to one bucket.
- To insert:

case 1: if bucket is full, and the bucket is not using all the indirection bits, the bucket can be split as follows:



Bucket 1 Full, we want to insert in bucket 1.

case 2: add a bit in indirection layer, and double all pointers. Then, you can insert as in case 1.

case: bucket not full, go ahead and insert data.