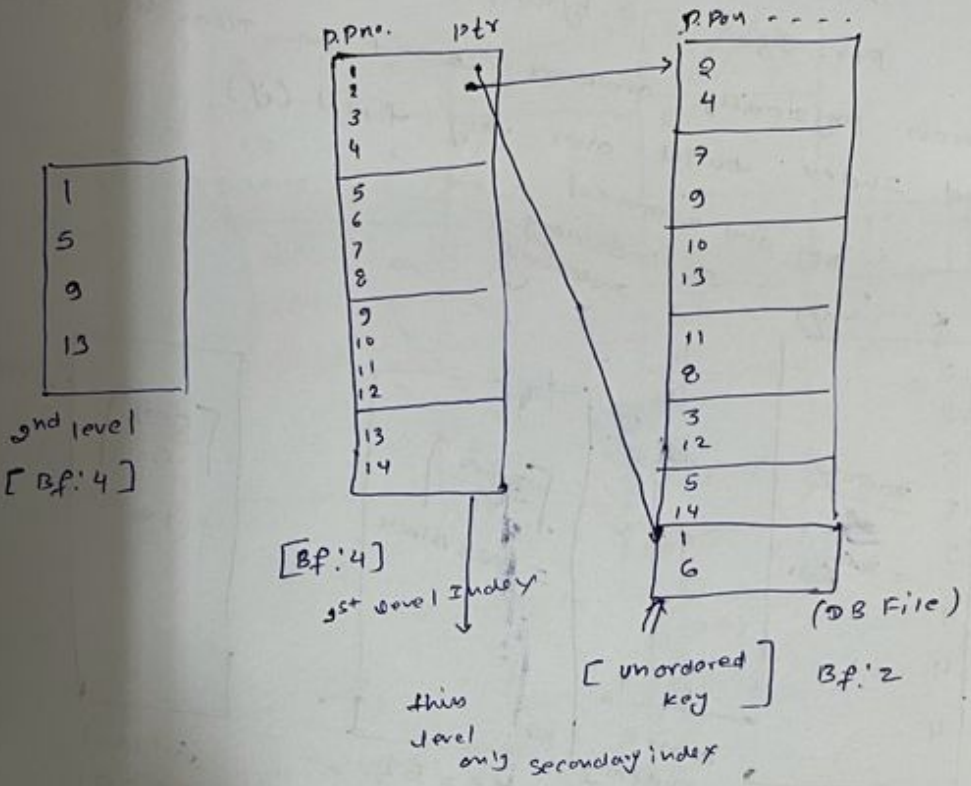
10/29/2021
Lec - 31

* Secondary Index.

(1) Search key used for index may not ordered field [un-ordered field] and

(2) Search key may be key (or) Non-key.



* Secondary index always dense index.
* More than one secondary index possible for DB Table.
* ~~Worst~~ worst case I/O cost :- (k+1) block.
* I/O cost of secondary Index more than I/O cost of Primary Index.

Q.

# of records 30000

Block size : 1024 Bytes [unspanned org. used]

Record size : 100 Bytes

Search key : 12 Bytes.

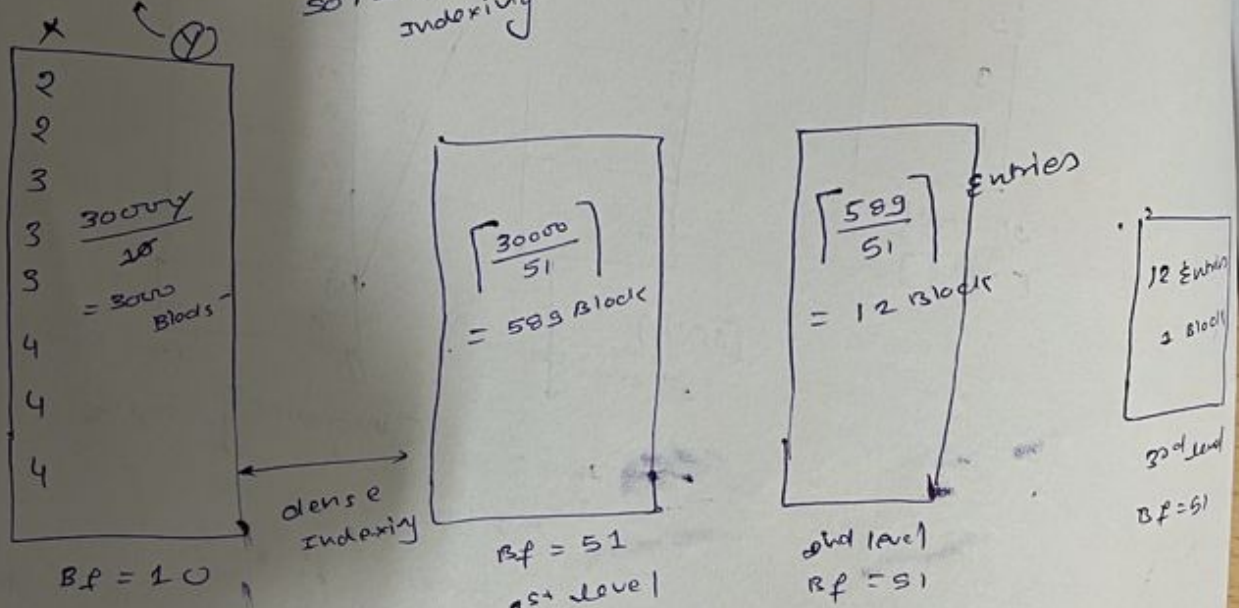
Ptr. size : 8 bytes.

Records physically ordered based on non-key field (x) and Index build over key field (y).

Solution

key and unordered

so, secondary Indexing



(a) Ans $k = 3$ level

(b) total Block = 589 + 12 + 1

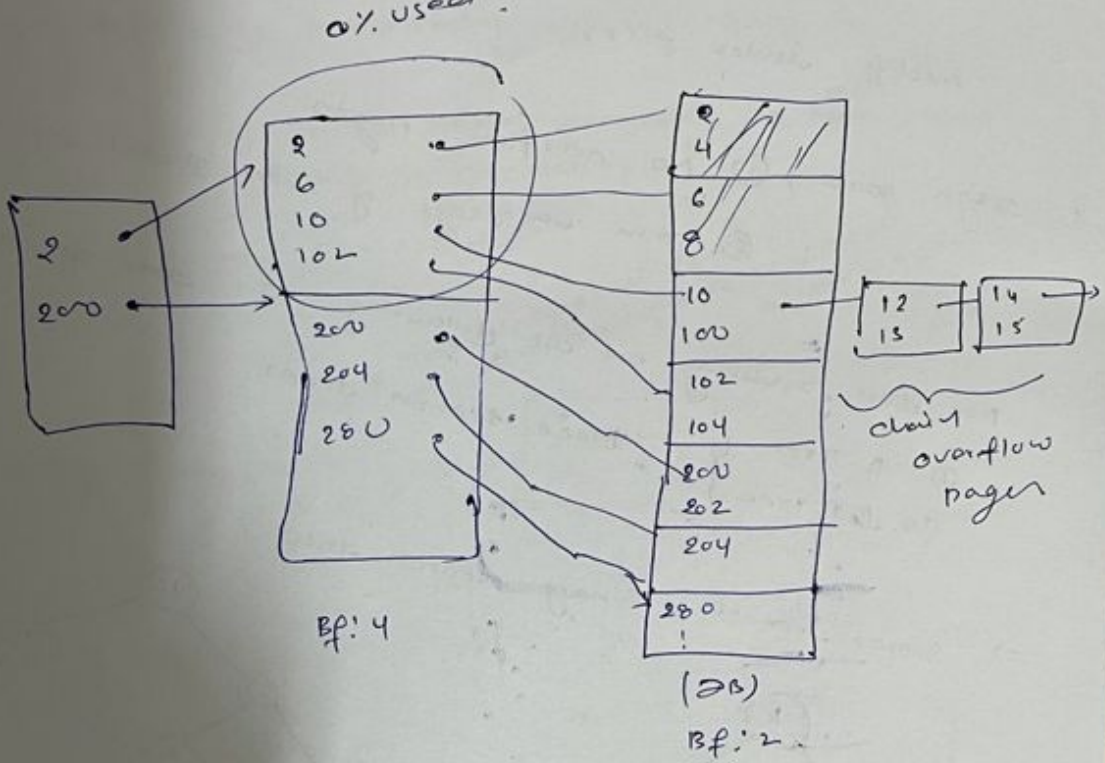
= 602

(3) I/O cost = k + 1

= 3 + 1

= 4 Block.

* Limitations of Static Multilevel Index:
(Row Indexing discussed before)



* Insertion ⇒ overflow pages
⇓
worst case I/O cost $O(n)$
n: # of DB Blocks

* Deletion ⇒ minimum usage of Index Block is ½.

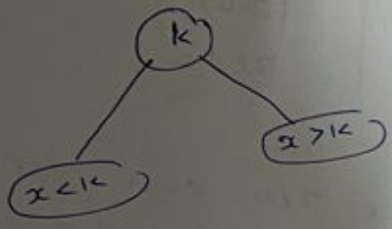
Because of these limitations we go through dynamic Multilevel Index.

* Dynamic Multilevel Index :- Based on insertion/deletion of data record need to modify index file.

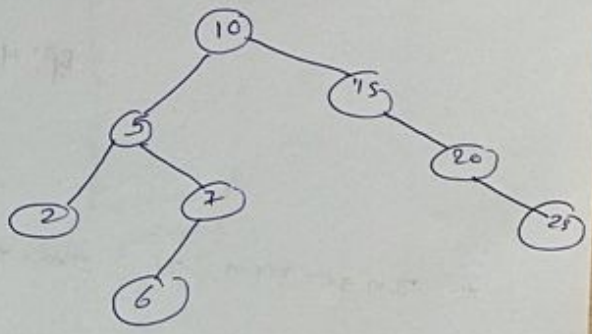
* Design goals :- ① No overflow pages
② min. usage of index should be 50%

For this Indexing we use following two data structures.
① B-Tree
② B+ Tree
} Blace search trees.

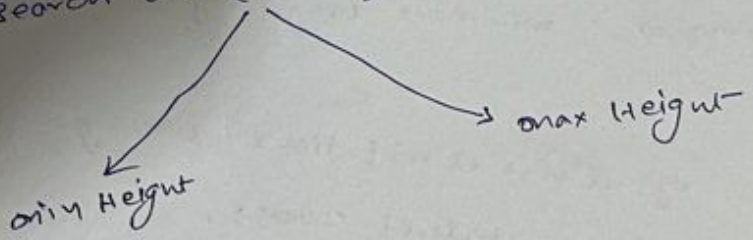
=> Search tree => Binary search tree:



or



=> Binary search tree (N - keys)



Query processing overview :

It is a procedure of transforming high-level query to low level language.

The Basic steps of query processing is as follows:

→ Parsing & translation

→ Optimization

→ Evaluation