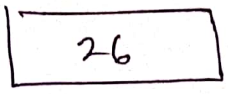
1705045

Iftekhar Hakim Kaowsar

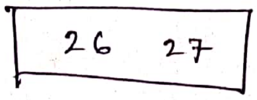
CT-2

Page-1

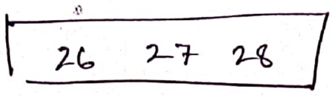
(1.) Insert 26 —



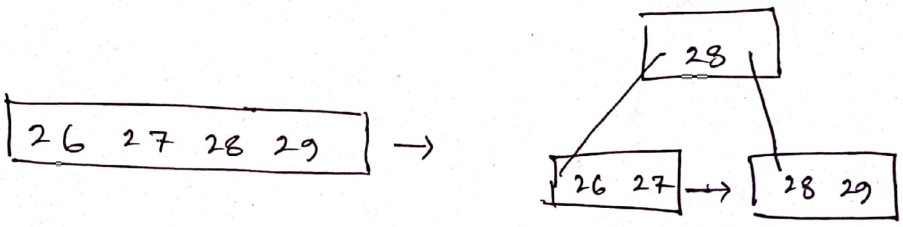
Insert 27



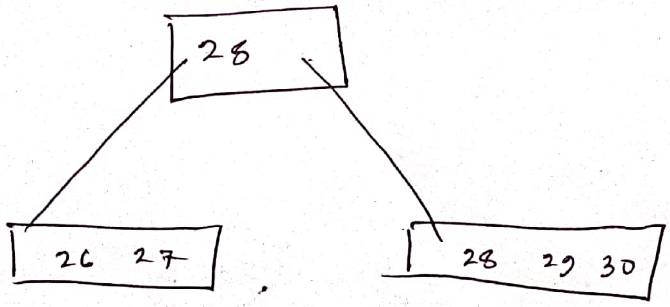
Insert 28



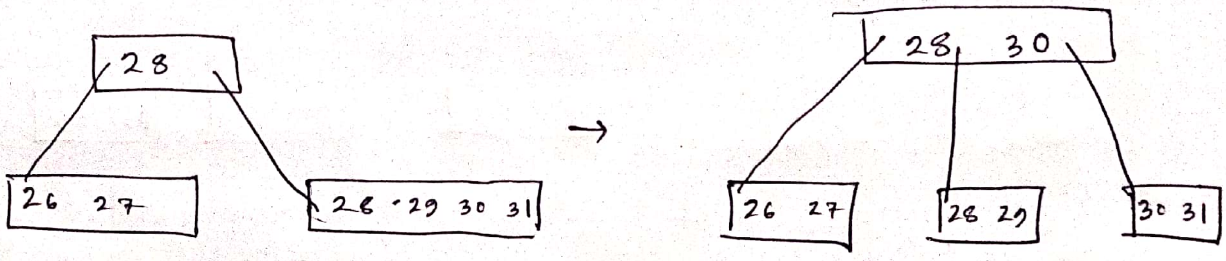
Insert 29



Insert 30

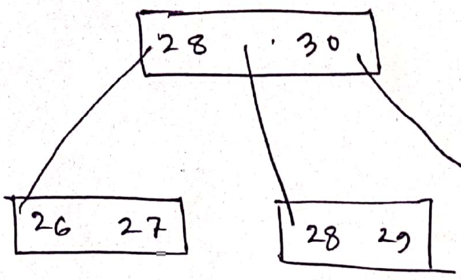


Insert 31

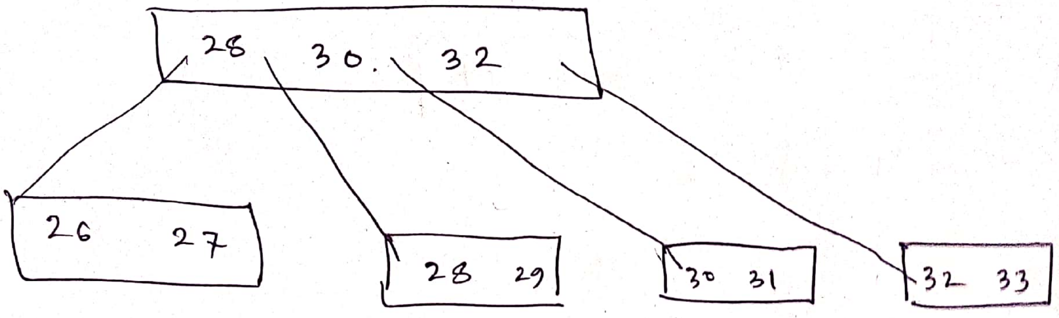


Roll- 1705045
Page-2

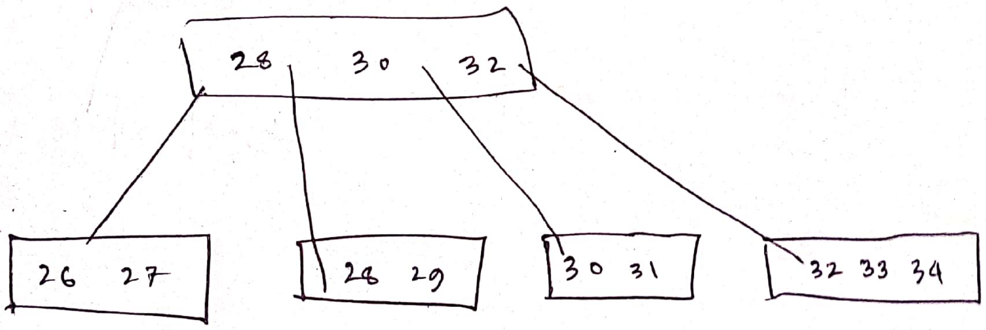
Insert 32



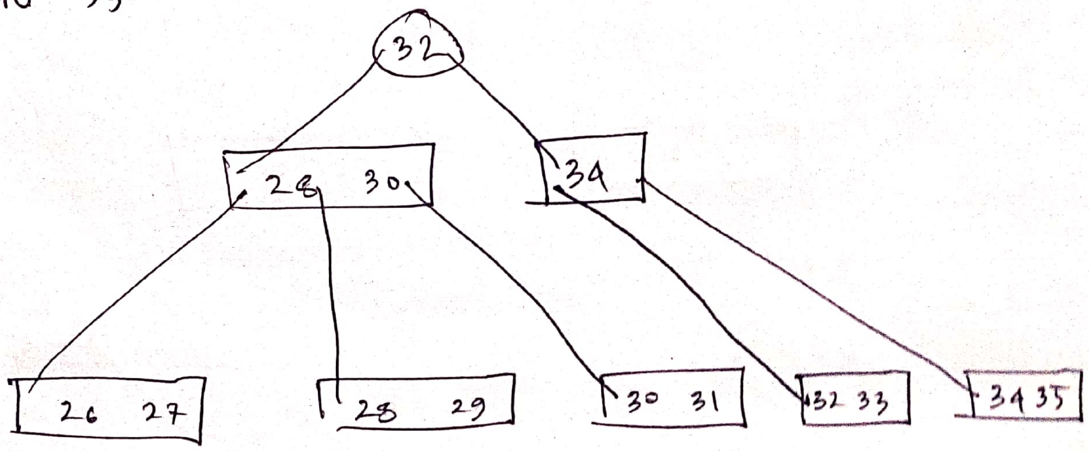
Insert 33



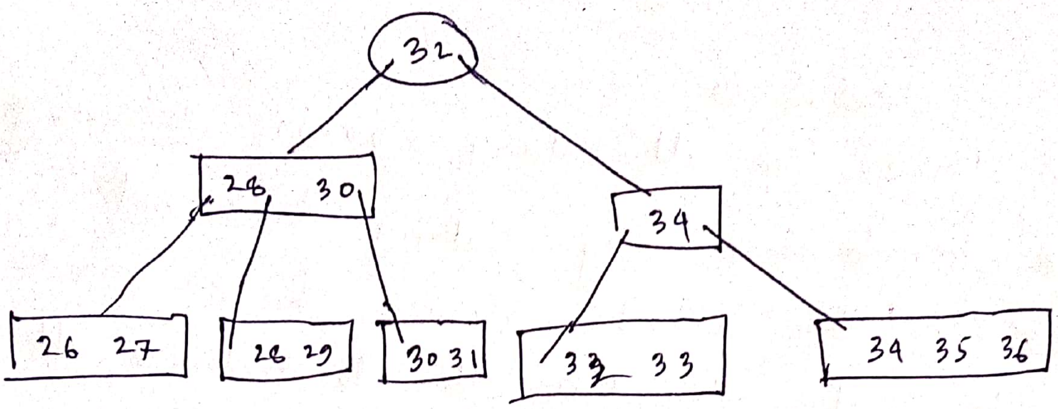
Insert 34



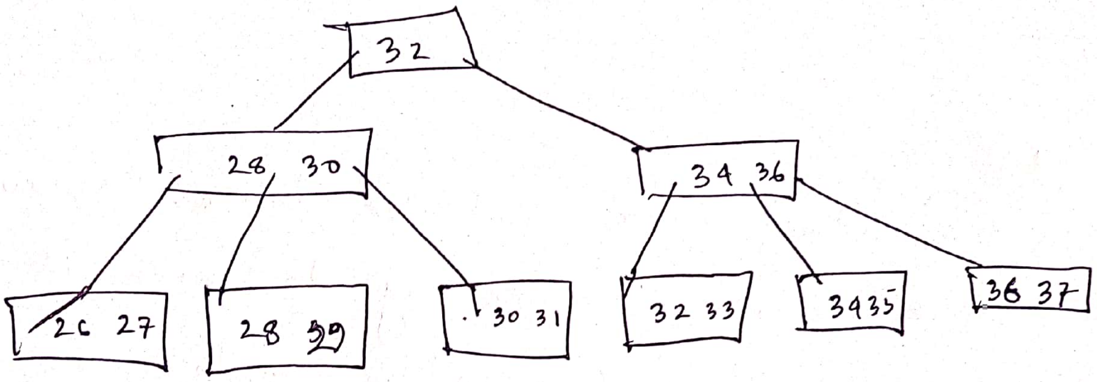
Insert 35



Insert 36:



Insert 37:



This is final.

(3.) Non-clustering index's sorted order does not match~~es~~ with actual order of records. That's why, if we ~~query~~ scan sequentially using a non-clustering index, we may have to access new block for every index. On magnetic disk sense, the head has to move from block to block every time (worst case). That's why, it is expensive on magnetic disk.

~~Select 4 f~~

Here for all salary ≤ 90000, ~~clustering~~ search key using salary does not mean the actual sequence of the main file. That's why ~~for all~~ to retrieve all records, we may have to select a block every time.

(4.)

Secondary index must be dense. Because there is no sensible way to make it sparse. In sparse index method, we select some keys and other keys are ~~defined~~ retrived by finding highest lower bound or lowest upper bound. As non-clustering index does not mean the actual order ~~of~~ or sequence how they are ~~sort~~ stored in main record file, we can not use the idea of sparse index here. That's why, it must be ~~non-~~ dense.

However, making it multi-level, we can make sparse indexing on second level.