• Exercise 4.1.1

The number of blocks we need to hold a data file is n/3.

1. A dense index: n/10 … total = n/3+n/10
2. A sparse index n/30 … total = n/3 + n/30

• Exercise 4.3.3

Let’s assume there’re n keys, thus the pointer is n+1. So, we have equations:

12\*n + 4 \* (n+1) <= 16384

The max value of n is 1023, thus there’re 1023 keys and 1024 pointers a block of 16384 bytes have.

12n+4(n+1) <= 16384

Max(n) = 1023

Thus, there are at most 1023 keys and 1024 pointers.

• Exercise 4.3.4

1. Interior Nodes: Minimum number of pointers are 6 and keys are 5.

Leaves: Minimum pointers are 5, and keys are 5.

1. Interior Nodes: Minimum number of pointers are 6 and keys are 5.

Leaves: Minimum pointers are 6, and keys are 6.

• Exercise 4.3.5

1. Lookup the record with key 41.

1) It will go to the 1st node and there is only one key 13, 41 is > 13 and hence it will go to Right child node.

2) Here we have 3 keys 23,31,43 and we know 41 will present in between 31 and 43 Hence it will go to 3rd child node.

3) Here it will perform a sequential search and there found the key 41.

4) This is a successful search.

1. Lookup the record with key 40.

1) It will go to the 1st node and there is only one key 13, 40 is > 13 and hence it will go to Right child node.

2) Here we have 3 kesy 23,31,43 and we know 40 will present in between 31 and 43, hence it will go to 3rd child node.

3) Here it will perform a sequential search but 40 is not there.

4) This is an unsuccessful search.

1. Lookup all records in the range 20 to 30.

        1) It will go to the 1st node and there is only one key 13, (20 to 30) is > 13

           and hence it will go to Right child node.

        2) Here we have 3 kesy 23,31,43 and we know 20 to 30 will present in between 23 and 31

            Hence it will go to 2nd child.

        3) Here it will have 23 and 29 in the range of 20 to 30.

        4) This is a successful search.

1. Lookup all records with keys less than 30.

        1) It will go to the 1st node and there is only one key 13, 13 is < 30

           and hence it will go to both Left and Right child nodes of 13.

        2) Here at the left side 7 is present and hence it will both left and right of 7

            and access all the nodes 2,3,5,7,11.

        3) Here at the right side <30 will be in the range 23 to 31 and hence it will

           access 13,17,19,23,29.

        4) This is a successful search.

1. Lookup all records with keys greater than 30.

        1) It will go to the 1st node and there is only one key 13, 13 is < 30

           and hence it will go to Right child nodes of 13.

        2) Here 31 and 43 is >30 and hence it will access all of its childern

            31,37,41,43,47.

       3) This is a successful search.

1. Insert a record with key 1.

       1 is < 13 and hence it will go to left child of 13 now

       1 is < 7 and hence it will go to left child of 7 now

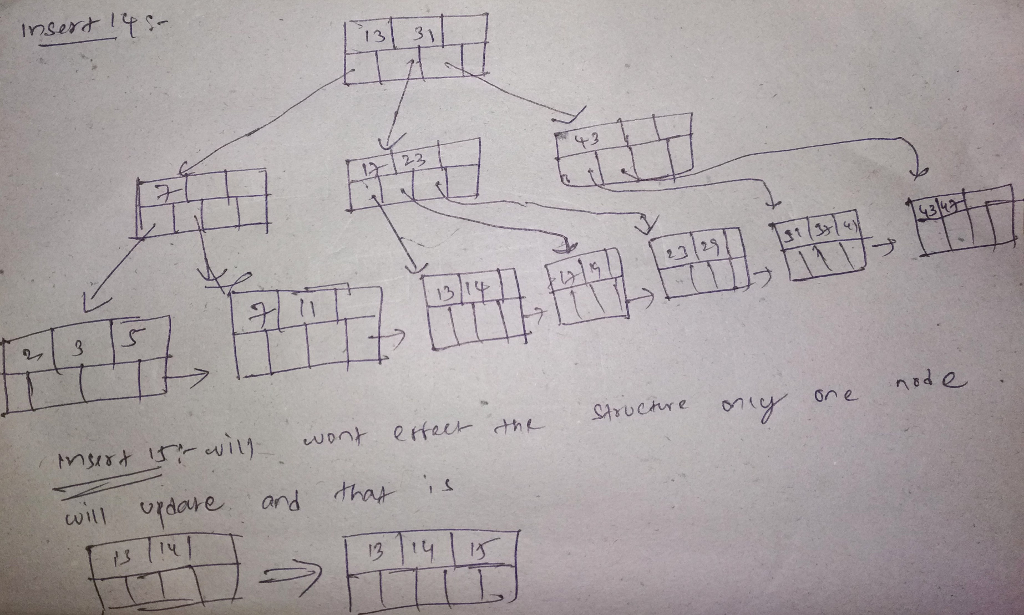
       Now 1 get in to its right place and the new node is 1 2 3 5

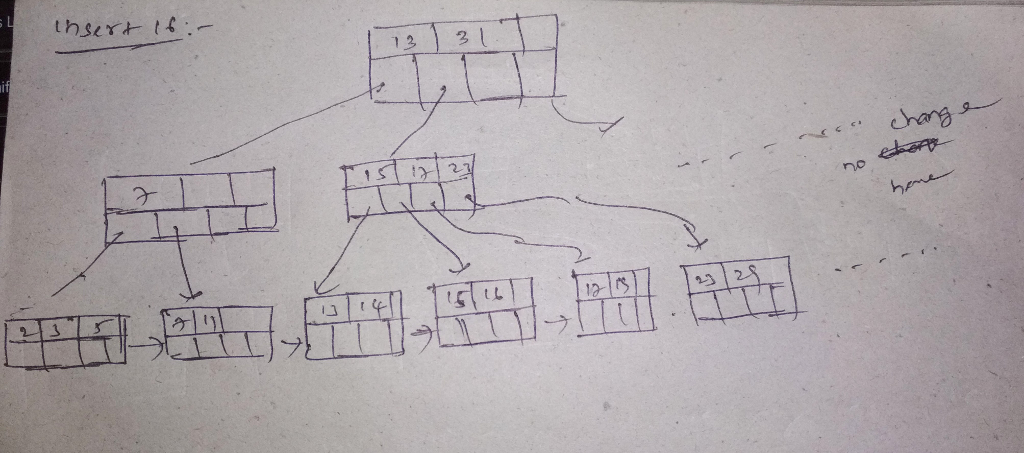
       but this is overflow hence and split will have

       3 will go to up and 1 2 will become left child of 3

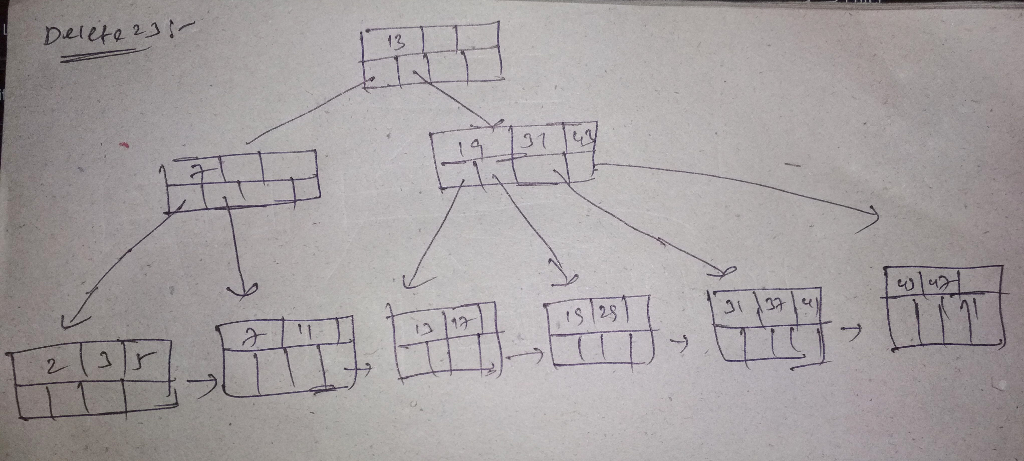
       and 5 will become rigth child of 3

1. Insert records with keys 14 through 16.

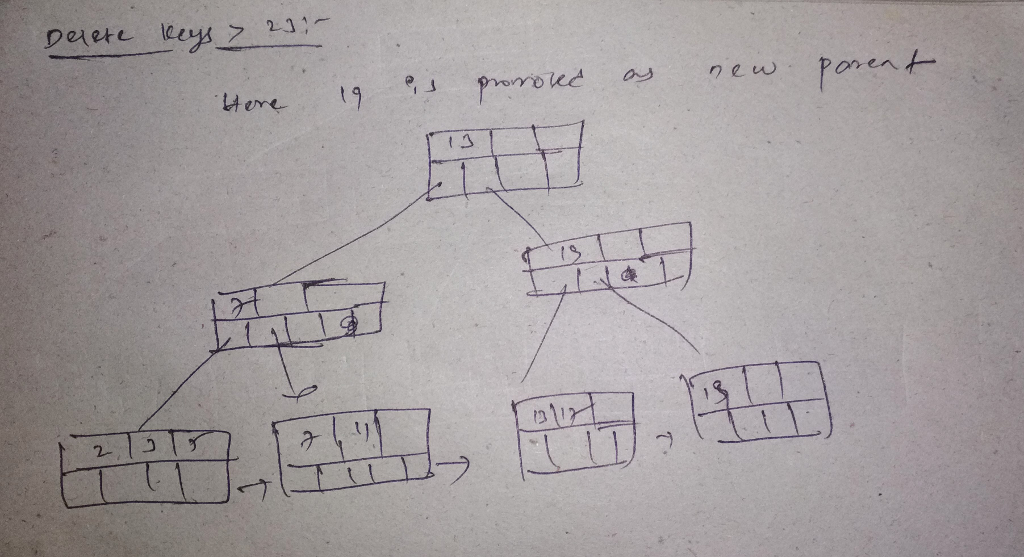




1. Delete the record with key 23.

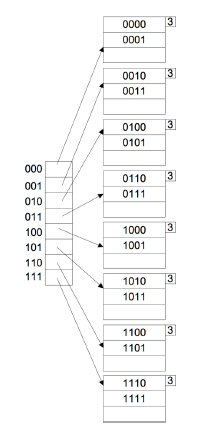


i) Delete all the records with keys 23 and higher

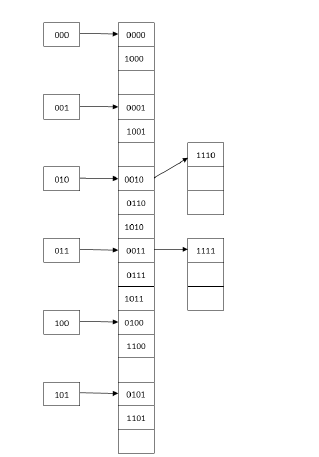


• Exercise 4.4.6

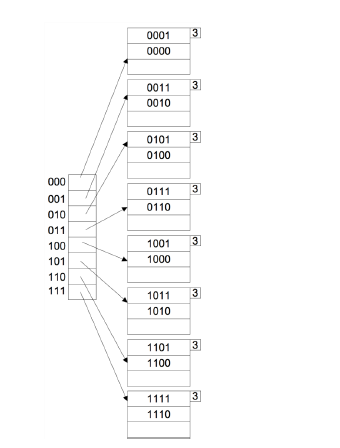
A



B



C



D

