Exercise

① 4.1.1.

1 block → 3 records
→ 10 pointers

dense records:-

for n records

# of blocks = $\frac{n}{3}$

for n pointers

# of blocks = $\frac{n}{10}$

∴ total = $\frac{n}{3} + \frac{n}{10} = \frac{10n+3n}{30} = \boxed{\frac{13n}{10}}$ //

Sparse records:

for n records

# of blocks = $\frac{n}{3}$ ⇒ same

But for each block there has to be one pointer and it can have only 3 records

∴ 10 * 3 = 30 pointer for ~~30~~ ~~block~~ total

$$\text{total} = \frac{n}{3} + \frac{n}{30}$$

$$= \frac{10n}{30} + \frac{n}{30}$$

$$= \boxed{\frac{11n}{30}} \text{ //}$$

② . 4.3.3

Pointers = 4 bytes Keys = 12 bytes

for any node there can be

n keys and (n+1) pointers

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

$$12n + 4n + 4$$

How many keys and pointers in 16384 bytes

$$16n + 4 = 16384$$

$$16n = 16380$$

$$n = \frac{16380}{16} = 1023.7$$

$$\boxed{n = 1023} \text{ //}$$

③ 4.3.4

Minimum number of keys and pointer in B-tree

(i) interior node

$n = 10$ 10 keys & 11 pointers

Interior ⇔ non-leaf node

∴ keys & ptrs

$n = 10$

$$\text{minimum ptrs} = \frac{n+1}{2} = \frac{10+1}{2} = \frac{11}{2} \text{ //}$$

$$\text{minimum keys} = \left[\frac{n+1}{2}\right] - 1 = \frac{10+1}{2} = \frac{10+1-2}{2} = \frac{9}{2} \text{ //}$$

$n = 11$

$$\text{Min. ptrs} = \frac{11+1}{2} = \frac{12}{2} = 6$$

$$\text{Min keys} = \left[\frac{11+1}{2}\right] - 1 = \frac{12}{2} - 1 = 6 - 1 = 5$$

(ii) Leaf node

n=10 Min. ptrs = $\frac{n+1}{2} = \frac{10+1}{2} = \frac{11}{2}$

Min keys = $\frac{n+1}{2} = \frac{10+1}{2} = \frac{11}{2}$

n=11 Min Ptrs = $\frac{n+1}{2} = \frac{11+1}{2} = \frac{12}{2} = 6$

Min keys = $\frac{n+1}{2} = \frac{11+1}{2} = \frac{12}{2} = 6$

4.3.6

4 bit sequences

blocks hold 3 records

Starts with 0 and 1

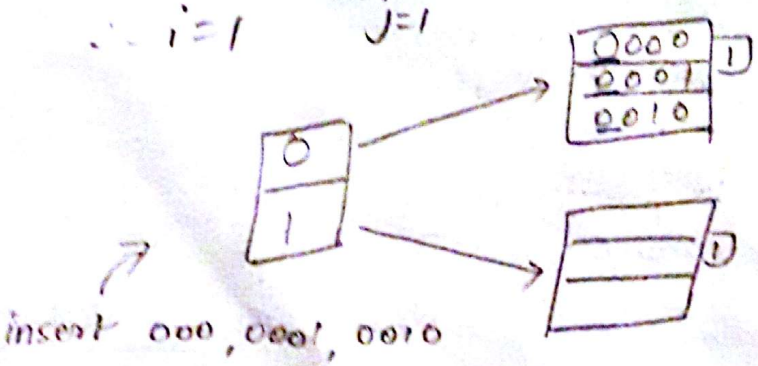
a) Insert 0000, ... 1111 method of hashing

extensible :

Starts with 0 and 1

Consider the first 'i' bits from the records

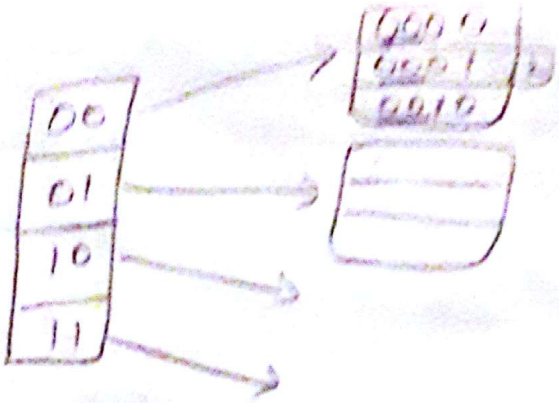
∴ i=1 j=1



insert 000, 0001, 0010

It overflows for 0010

∴ increase the bits of 'i'



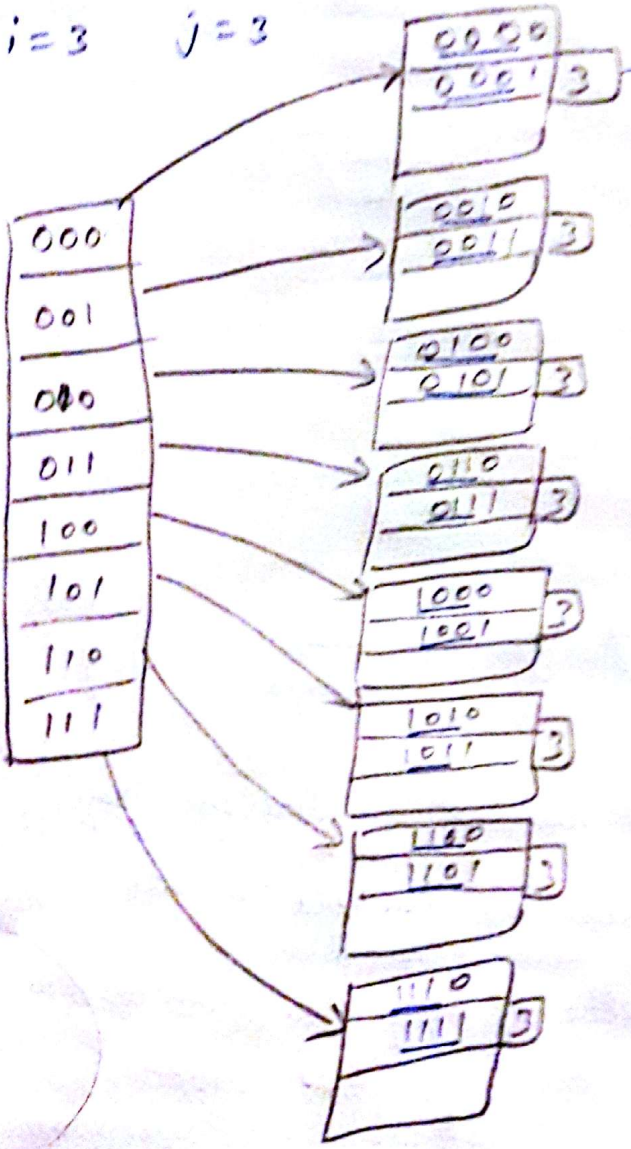
now i = 2

∴ j = 2.

* it again overflows for 0011

∴ increase the bits for i

∴ i = 3 j = 3



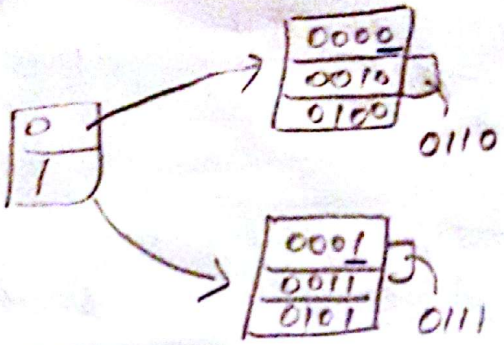
Inserting 0000 ..... 1111 into Extendible hashing method.

b) Linear hashing:

threshhold = 100 %.

7 = 100 %.

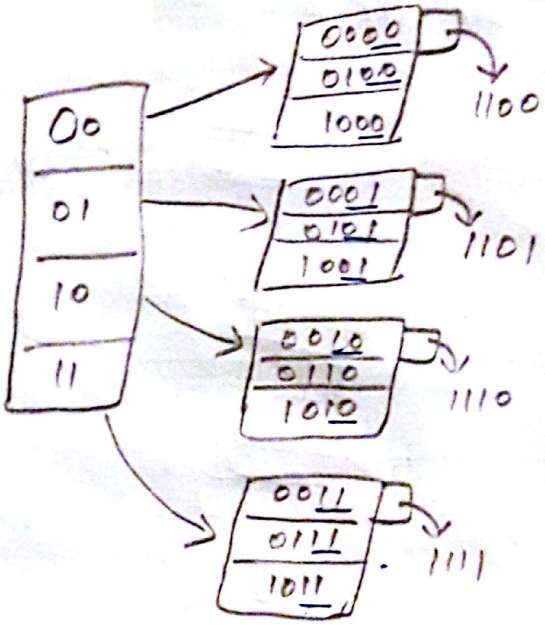
Insert from 0000 .... 1111.



for linear hash Consider the last bit

overflow occurs when we try to insert 1000

therefore increase the bits of n



$$\text{average occupancy} = \frac{r}{n \times f}$$

r = total number of search keys

n = number of buckets in use

f = number of keys that can be stored in one block

$r = 16$

$n = 4$

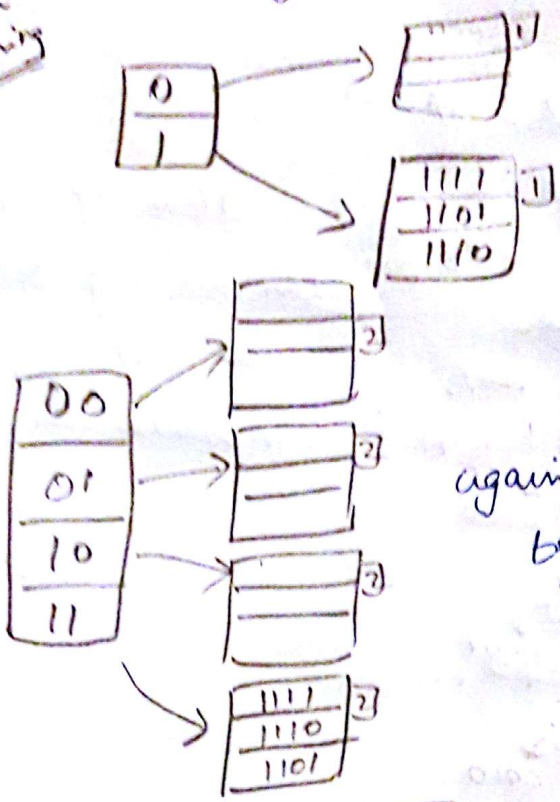
$f = 4$

$$\therefore = \frac{16}{4 \times 3} = \frac{16}{12} = 1.33 > 100\%$$

c) Insert 1111 0000

Extensible Hashing

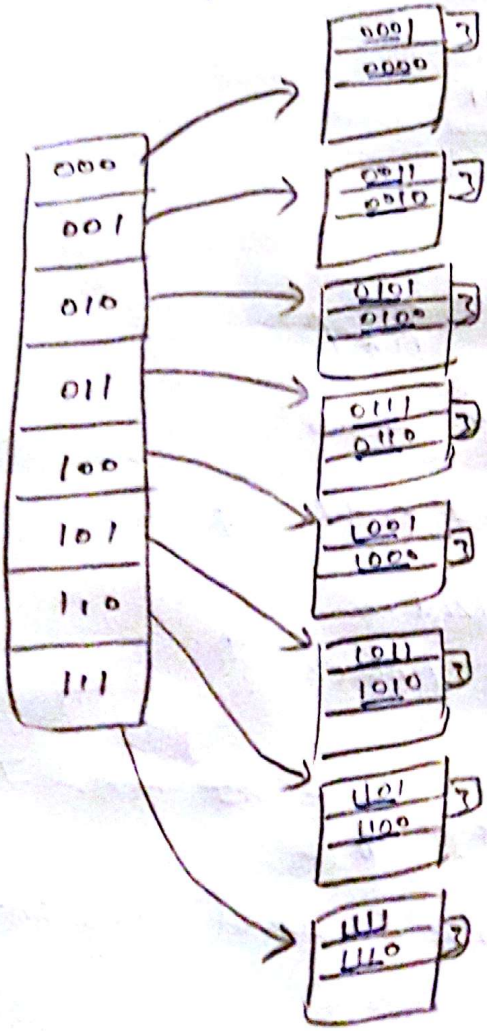
i = 1 j = 1



overflow occurs when we try to insert ~~1001~~ 1100 therefor increas the number of bits

again overflow occur when we try to insert 1100

∴ increas the i again



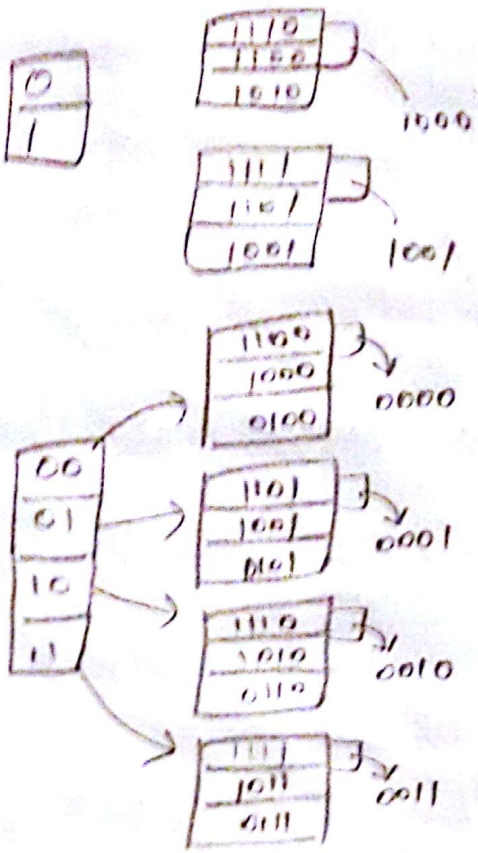
i = 3 j = 3

Without overflow 1111 .... 0000 has been inserted.

d) 1111 0000

linear hashing

threshhold 7 = 75%.



overflow occurs when we try to insert 1000 0111

∴ increase the bits

from above case

average occupancy = $\frac{r}{n * i}$

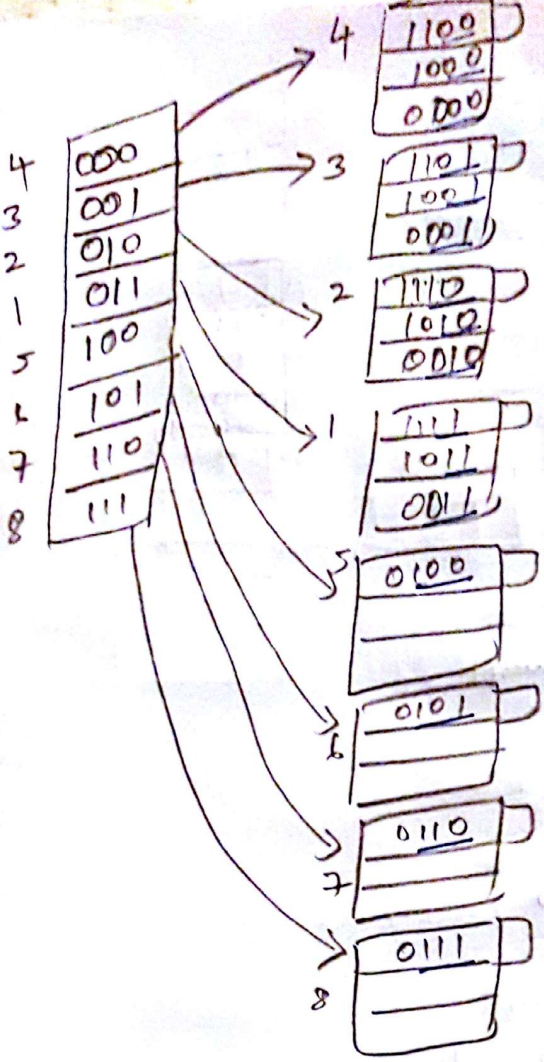
r = total no of keys = 16

n = number of buckets in use = 4

i = keys can be stored in 1 block = 4

$= \frac{16}{4 * 3} = \frac{16}{12} = 1.33$ ~~1.333 1.00~~ > 75%.

Since the threshhold level went above 75%.



$\therefore$ average occupancy $= \frac{r}{n * s}$

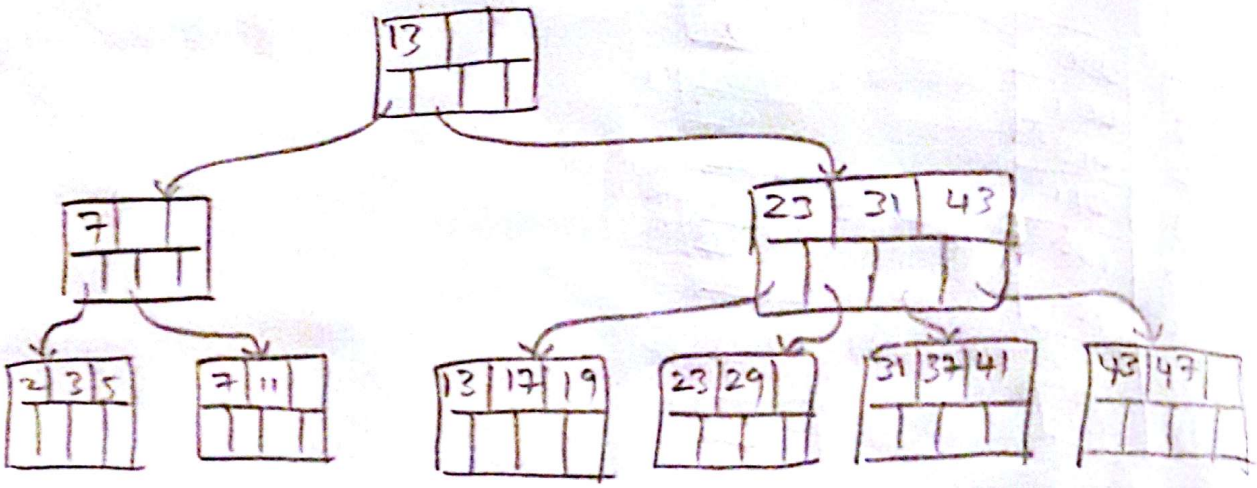
r = total keys = 16

n = current no of buckets in use = 8

s = Max no of keys in a bucket = 3

$\therefore = \frac{16}{24} = 0.66 < 0.75$

Ex: 4.3.5 Describe the changes for operations that modify the tree.



a) lookup the record with key 41

$13 \longrightarrow 31 \longrightarrow 41$

b) lookup the record key 40

$13 \rightarrow 31 \longrightarrow X$

record not found.

c) Lookup all records in range 20 to 30

(i) 23

(ii) 29 $13 \rightarrow 23$, $13 \rightarrow 31 \rightarrow 23$

$13 \rightarrow 23 \rightarrow 29$, $13 \rightarrow 31 \rightarrow 29$

d) look all key less than 30

(i) 2 $13 \rightarrow 7 \rightarrow 2$

(ii) 3 $13 \rightarrow 7 \rightarrow 3$

(iii) 5 $13 \rightarrow 7 \rightarrow 5$

(iv) 7 $13 \rightarrow 7$

(v) 11 $13 \rightarrow 7 \rightarrow 11$

(vi) 13 13

(vii) 17 $13 \rightarrow 23 \rightarrow 17$

(viii) 19 $13 \rightarrow 23 \rightarrow 19$

(ix) 23 $13 \rightarrow 23$, $13 \rightarrow 31 \rightarrow 23$

(x) 29 $13 \rightarrow 23 \rightarrow 29$, $13 \rightarrow 31 \rightarrow 29$

e) look all records greater than 30

(i) 31 — 13 → 31 , 13 → 43 → 31

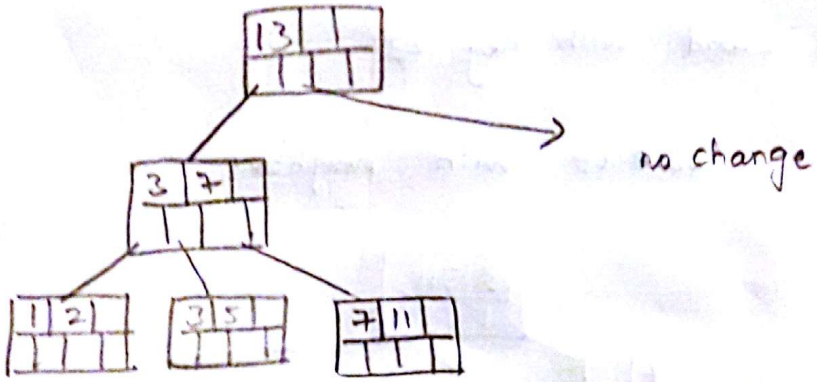
(ii) 37 — 13 → 31 → 37 , 13 → 43 → 37

(iii) 41 — 13 → 31 → 41 , 13 → 43 → 41

(iv) 43 — 13 → 43

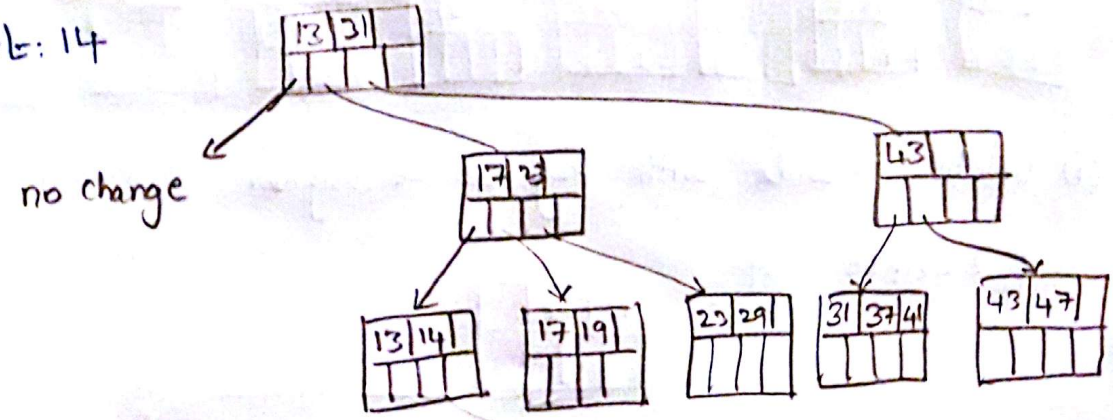
(v) 47 — 13 → 43 → 47.

f) Insert record with key 1

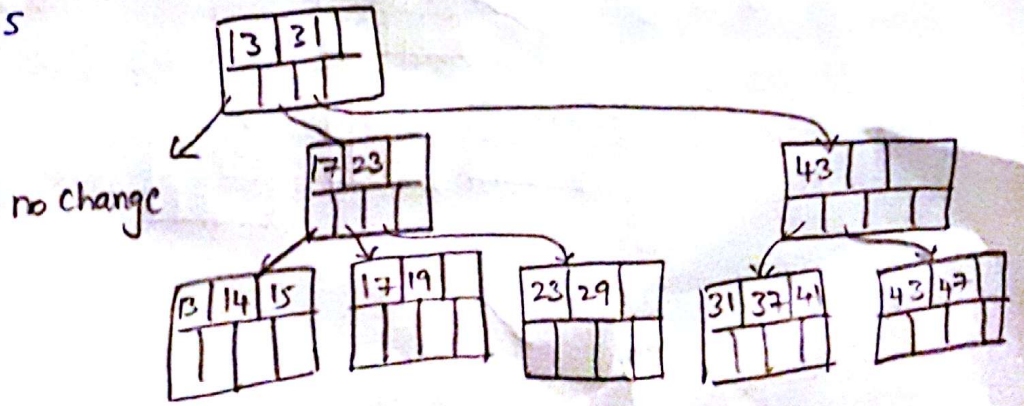


g) Insert records with key 14 through 16

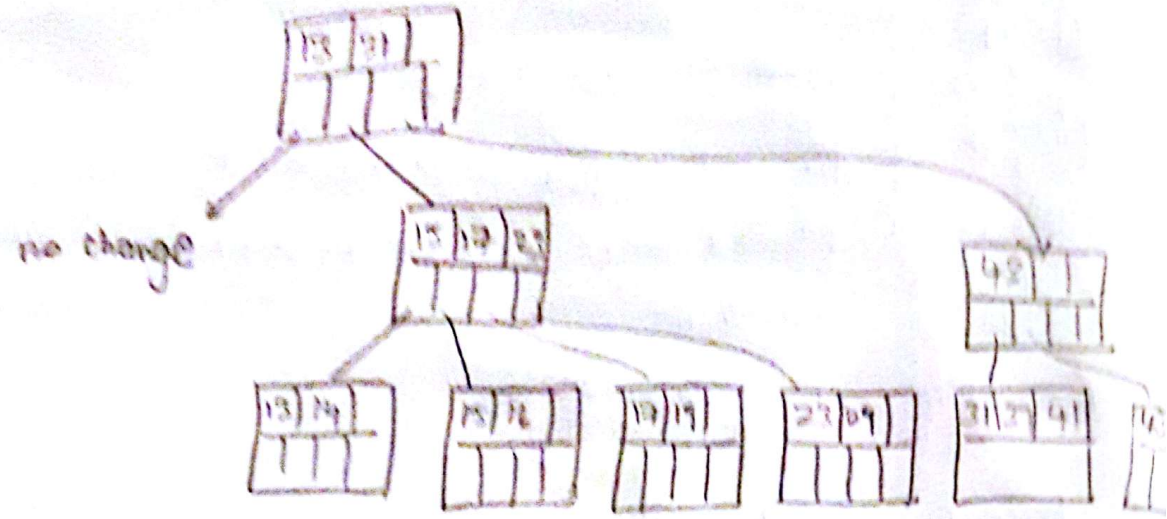
Insert: 14



Insert : 15

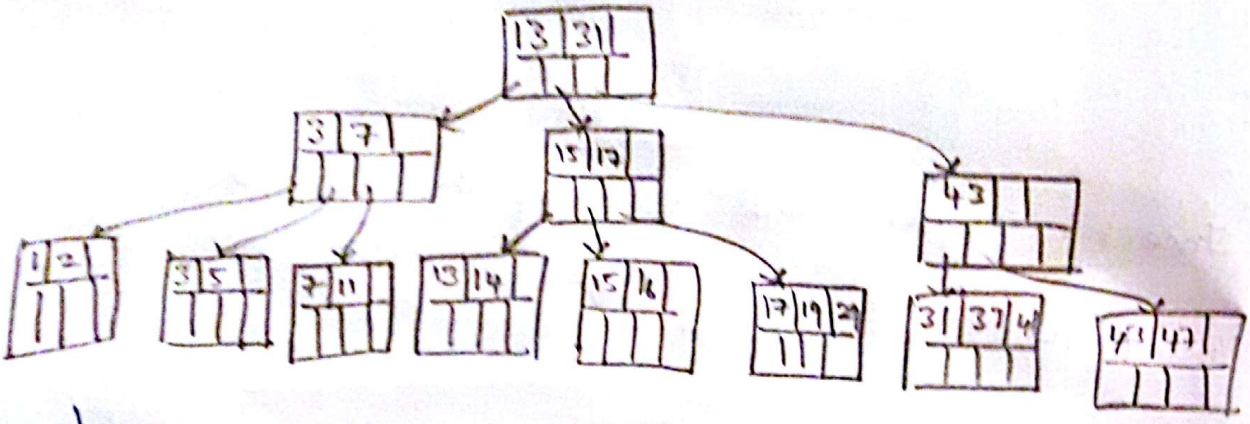


Insert : 16



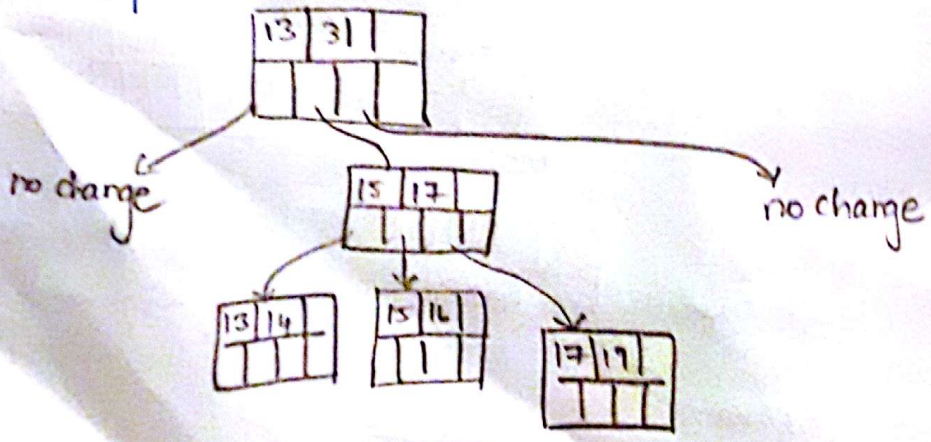
(h) Delete record with key 23

Coalesce with neighbor

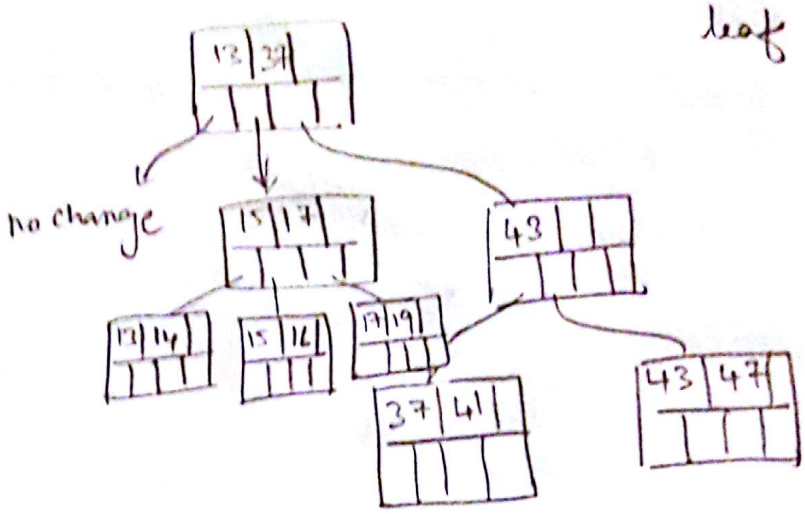


(i) Delete records with key 23 & higher

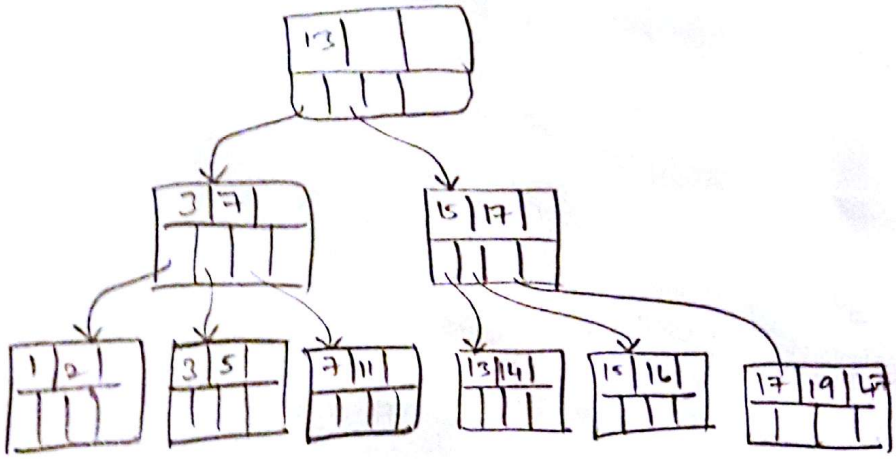
(i) Delete 29



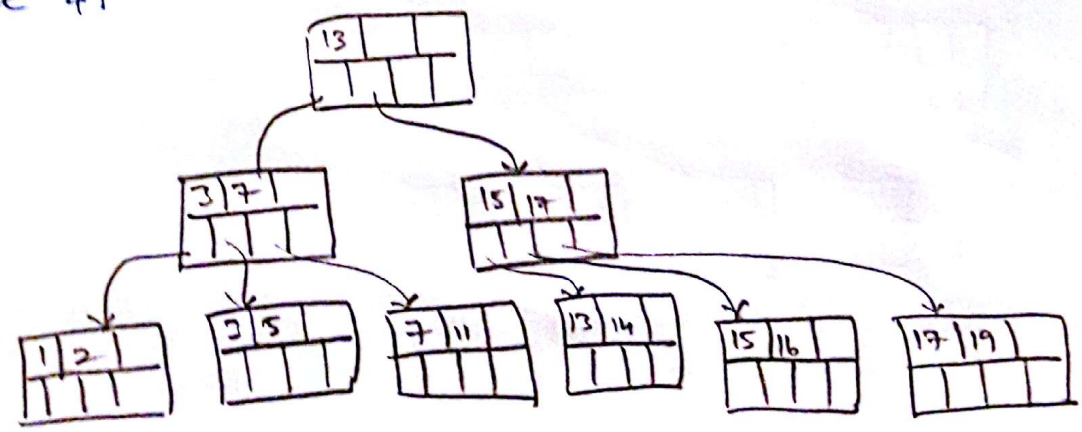
(ii) Delete 31 ← 37 came to root from leaf



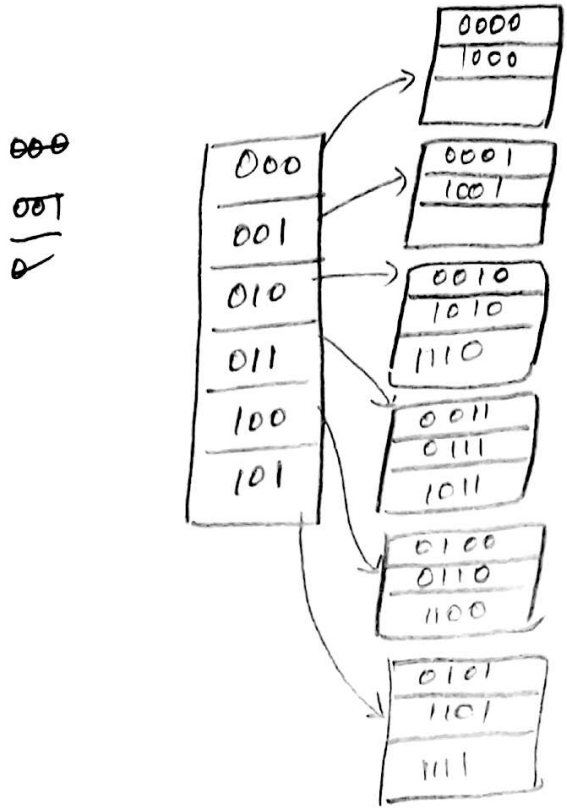
(iii) Delete 37, 41, 43



(iv) Delete 47



So we have to increase the number of buckets



$$\text{average occupancy} = \frac{r}{n \times r}$$

total keys = $r = 16$

n = number of buckets = 6

r = keys stored = 3

$$= \frac{16}{18} = 0.88 < 100\%.$$