ADB Assignment # 02

Q1-

B = 4096 bytes
r = 10,000,000
R = 150 bytes
75000 records per department
Primary key :- ROLLNO

(a) Select * from Student where ROLLNO = 9187
(file not ordered)

As file is not ordered, so linear search will be used.

$$Bfr = \left\lfloor \frac{B}{R} \right\rfloor = 27$$

$$b = \left\lceil \frac{r}{Bfr} \right\rceil = 370371$$

So number of block fetches = b = 370371

(b) As search is on ROLLNO and file is also ordered on ROLLNO which is primary key (unique column).
So binary search used.

number of block fetches = $\lceil \log_2 b \rceil = 19$

(c) As file is not ordered, so linear search.

Number of block fetches = b = 370371

(d) As search is on deptNo and file is ordered on deptNo but it is not unique column. So,

$$\text{Blocks to be fetched} = \left\lceil \log_2(b) + \frac{s}{Bfr} - 1 \right\rceil$$

where s is selectivity. = 75000

$$\text{Blocks} = \left\lceil \log_2(370371) + \frac{75000}{27} - 1 \right\rceil$$

Blocks to be fetched = 2796

Qx- Customer ID values:-

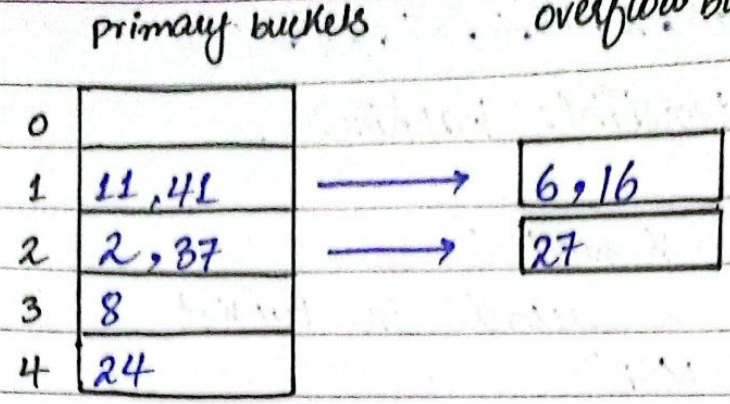
{11, 24, 8, 2, 41, 6, 37, 27, 16}

(a) 5 buckets

max 2 records in bucket

$h(K) = K \bmod 5$

loading records.



11 mod 5 = 1
24 mod 5 = 4
8 mod 5 = 3
2 mod 5 = 2
41 mod 5 = 1
6 mod 5 = 1 (overflow)
37 mod 5 = 2
27 mod 5 = 2 (overflow)
16 mod 5 = 1 (overflow bucket)

(b) Average no. of block access.

For 11 :- 1 block access
For 24 :- 1 block access
For 8 :- 1 block access
For 2 :- 1 block access
For 41 :- 1 block access
For 6 :- 2 blocks access
For 16 :- 2 blocks access
For 37 :- 1 block access
For 27 :- 2 blocks access
So Total = 12
So,
average blocks access = $\frac{12}{9} = 1.33$

11, 24, 8, 2, 41, 6, 37, 27, 16

(c) Extendible hashing

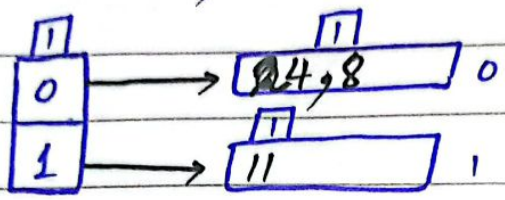
$h(K) = K \bmod 6$

max 2 records in bucket

(use MSB)

initially $d=0$, $d'=0$

then,



11 mod 6 = 101

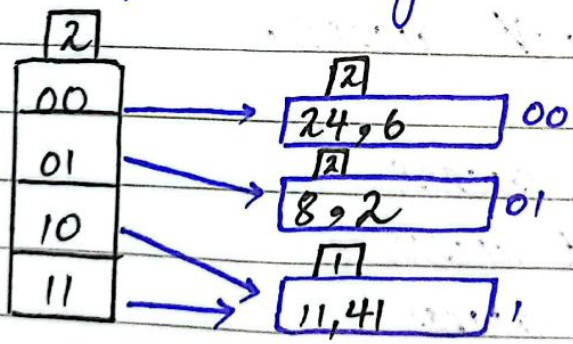
24 mod 6 = 000

8 mod 6 = 010

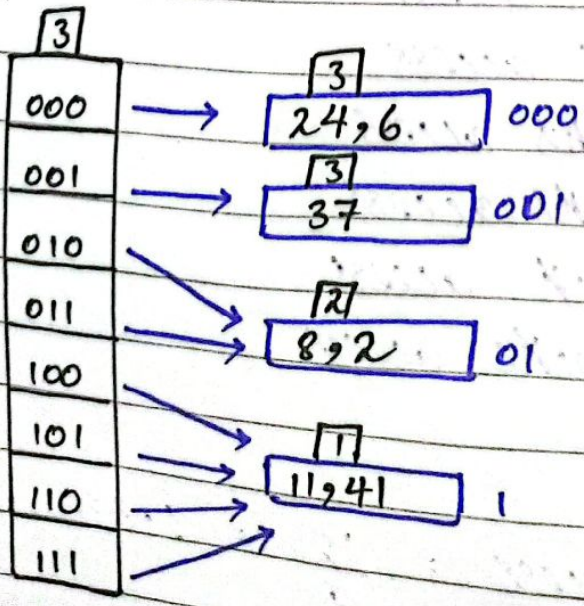
2 mod 6 = 010

(overflow, so split bucket 0)

Double directory size as local depth is equal to global depth



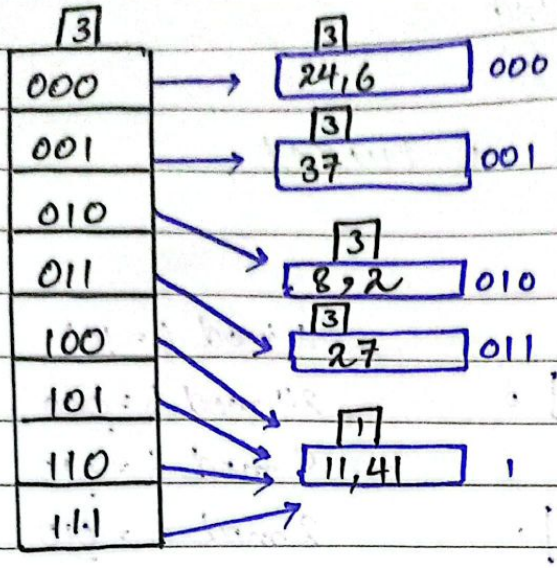
41 mod 6 = 101



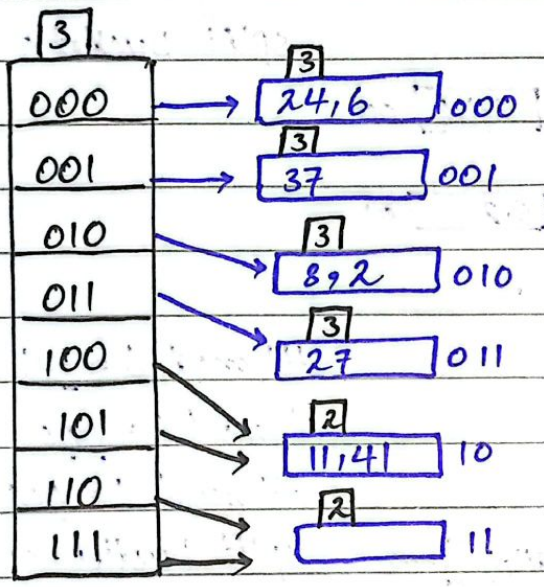
6 mod 6 = 000

37 mod 6 = 001

so: overflow, split bucket 00 & double directory as local & global Depth same.

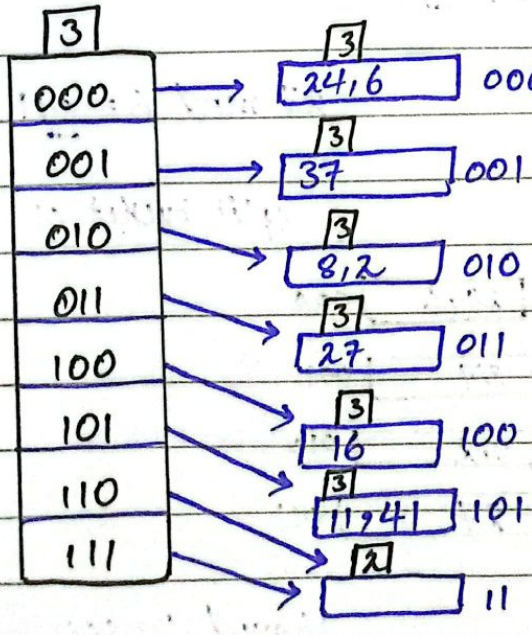


27 mod 6 = 011
so split bucket 01
but no need to
double directory
as local depth = 2 <
global depth



16 mod 6 = 100
overflow split bucket
10 but don't double
directory

Still 16 cause
overflow in bucket
10, again split



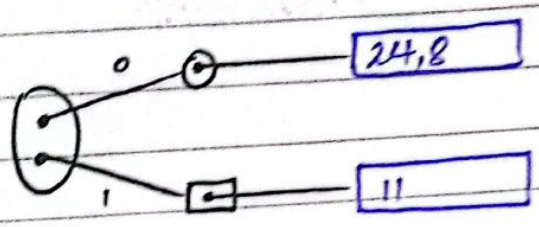
11, 24, 8, 2, 41, 6, 37, 27, 16

(d) Dynamic Hashing

$h(K) = K \bmod 6$

max 2 records in bucket.

Use MSB



11 mod 6 = 101

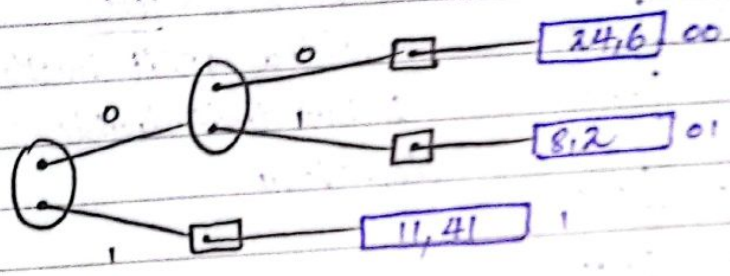
24 mod 6 = 000

8 mod 6 = 010

2 mod 6 = 010

overflow.

split bucket 0.

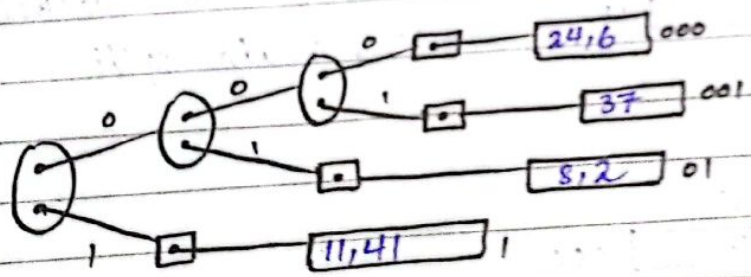


41 mod 6 = 101

6 mod 6 = 000

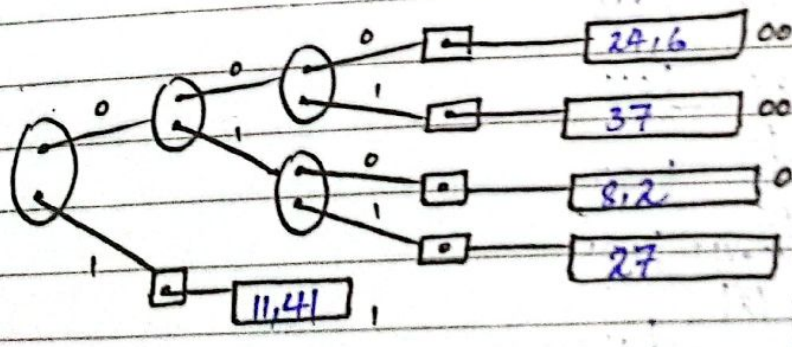
37 mod 6 = 001

split bucket 00



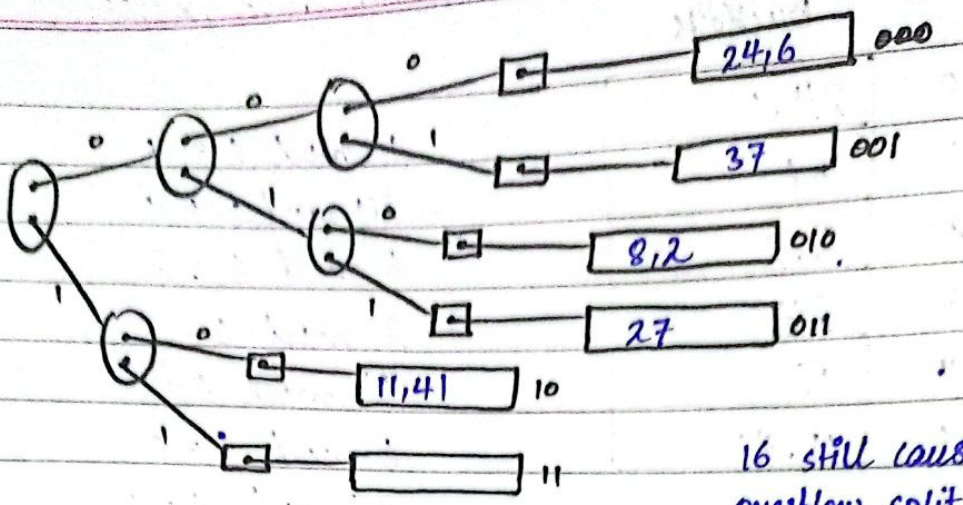
27 mod 6 = 011

split bucket 01

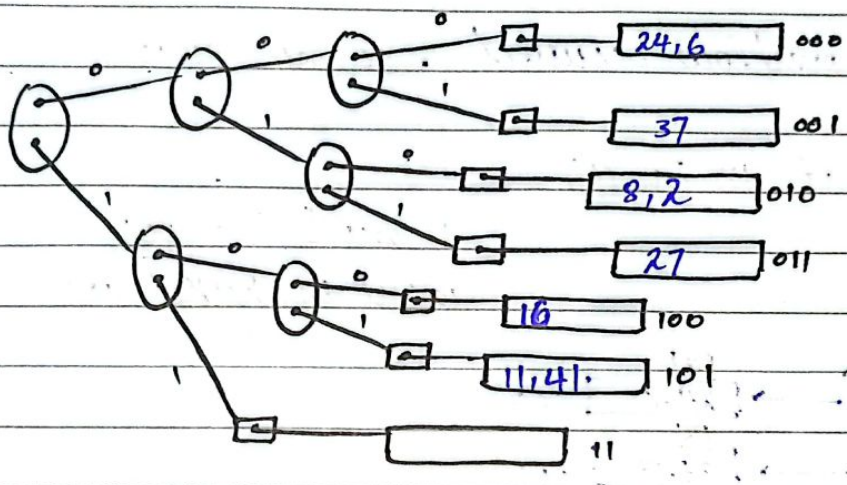


16 mod 6 = 100

split bucket 1



16 still cause overflow split bucket 10



(c) Linear Hashing

2 max records per bucket

$h(K) = K \bmod 6$

Initially we will have 6 buckets (0-5)

If overflow occurs, put key in overflow bucket and increment ~~double~~ the number of buckets as new hash function = $K \bmod 2M$ and split bucket pointed by n using new hash function

if $h_1(K) \neq n$ (use initial hash function)

if $h_1(K) < n$ (use new hash function)

n 0

11, 24, 8, 2, 41, 6, 37, 27, 16

primary Buckets

0	24, 6
1	37
2	8, 2
3	27
4	16
5	11, 41

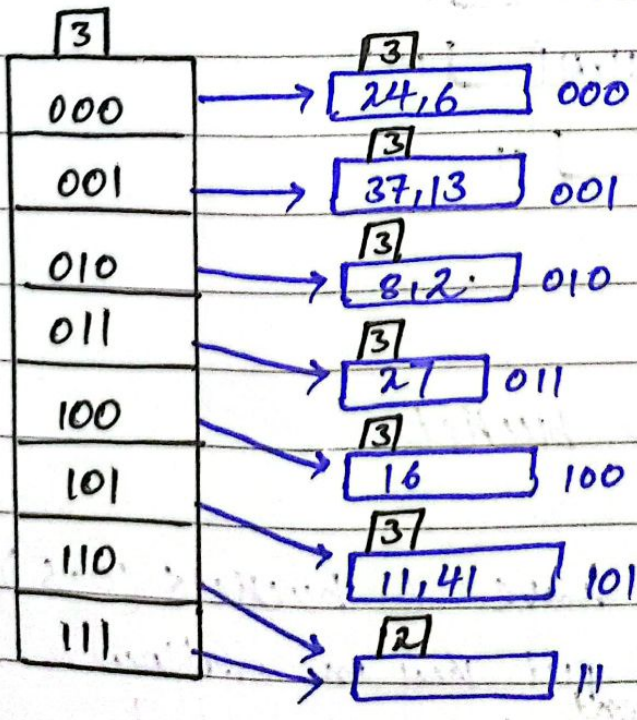
11 mod 6 = 5
24 mod 6 = 0
8 mod 6 = 2
2 mod 6 = 2
41 mod 6 = 5
6 mod 6 = 0
37 mod 6 = 1
27 mod 6 = 3
16 mod 6 = 4

No overflow occurred, so no split

(b) Insert 13 in (c)

13 mod 6 = 001

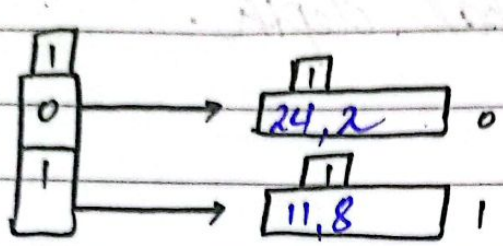
insert in bucket 001



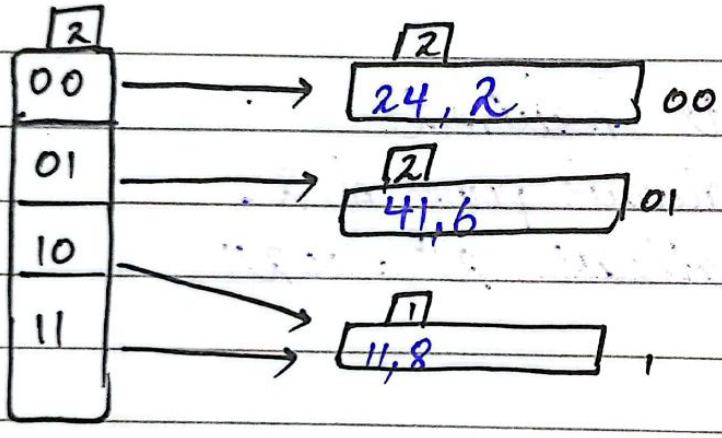
(g) Original Solution

As 18 continue to cause overflow in bucket 0000 everytime so change hash function to K mod 12 (rehash & reallocate all keys)

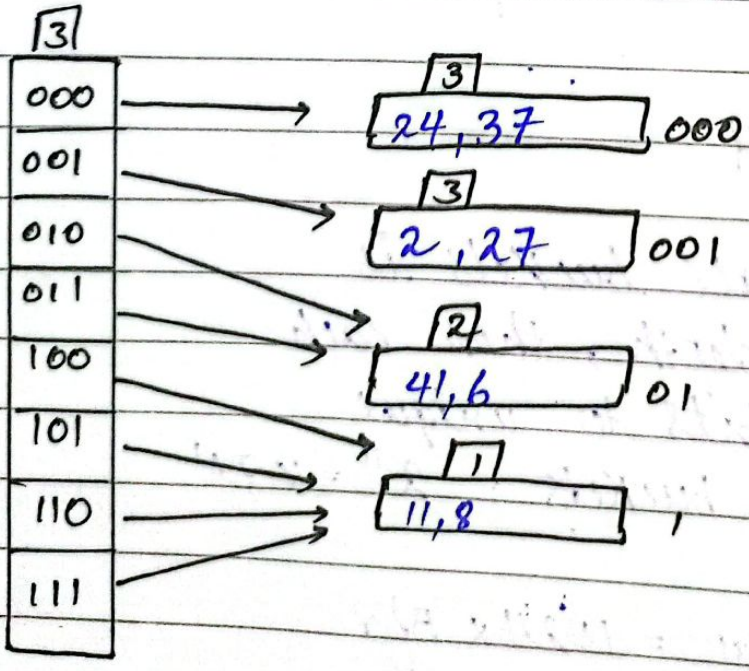
To insert :- 11, 24, 8, 2, 41, 6, 37, 27, 16, 13, 18



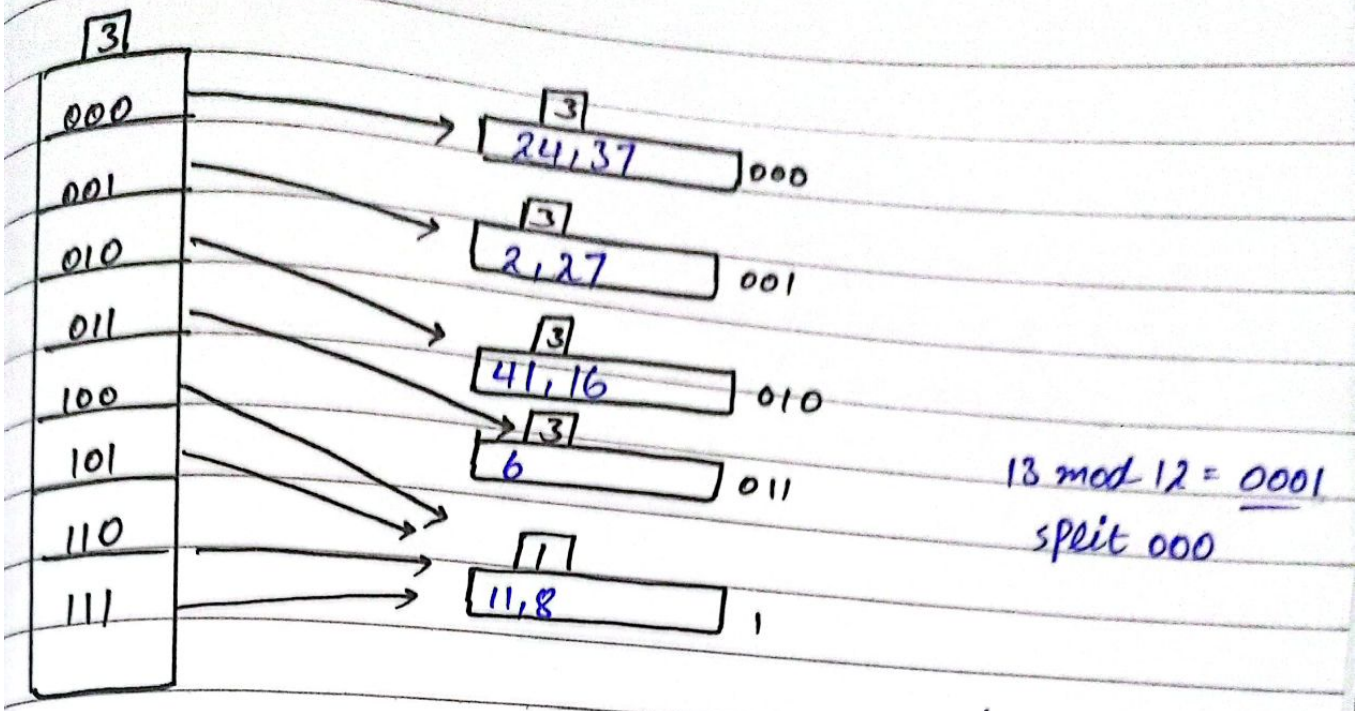
11 mod 12 = 1011
24 mod 12 = 0000
8 mod 12 = 1000
2 mod 12 = 0010
41 mod 12 = 0101
split 0



6 mod 12 = 0110
37 mod 12 = 0001
split 00

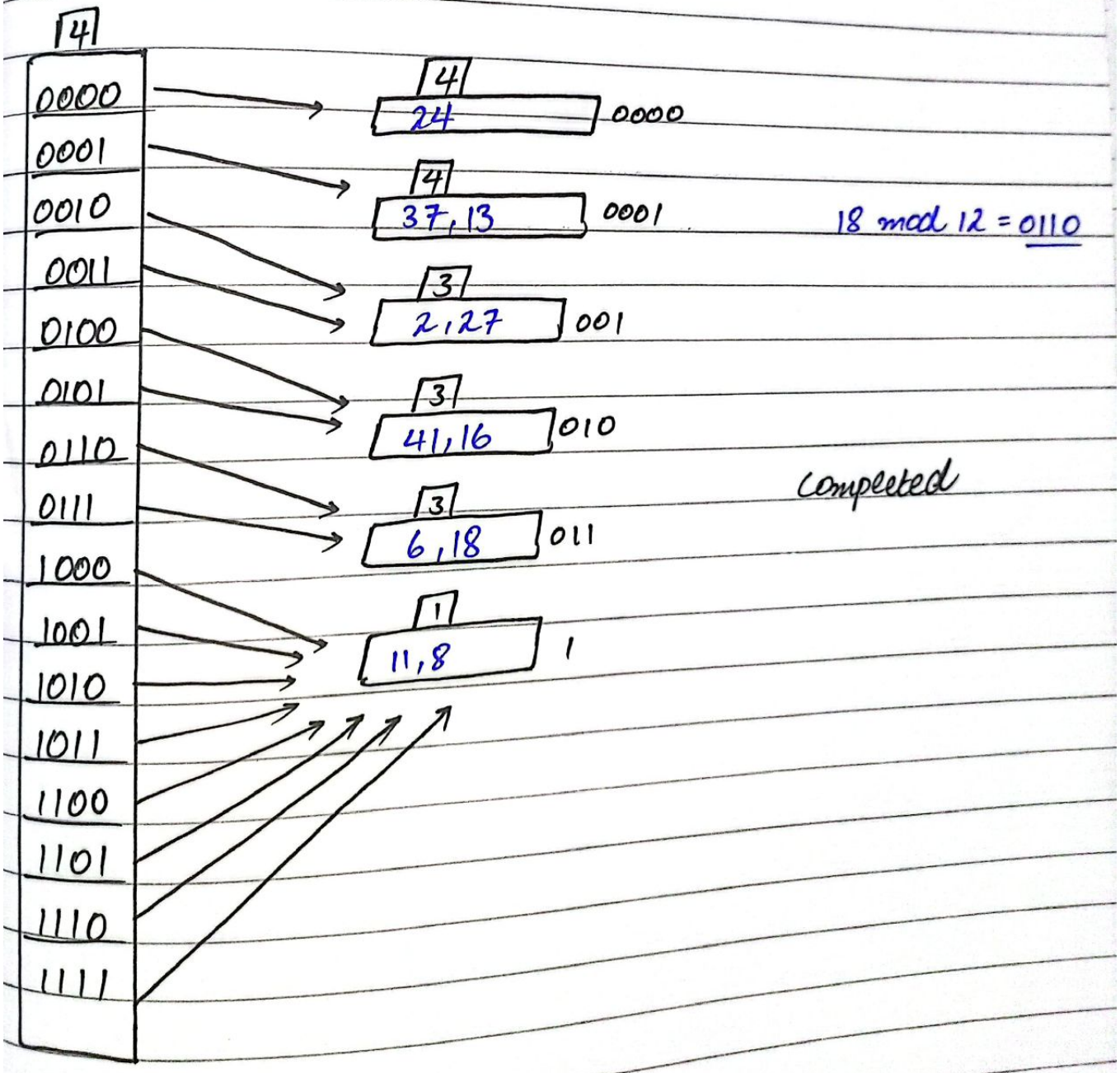


27 mod 12 = 0011
16 mod 12 = 0100
split 01



13 mod 12 = 0001

split 000



18 mod 12 = 0110

completed

Q3 Extendible hash.

r = 100,000
R = 16 + 16 = 32 bytes
B = 4096 bytes

(a) lowest possible global depth.

$Bfr = \lfloor \frac{B}{R} \rfloor = 128$

$b = \lceil \frac{r}{Bfr} \rceil = 782$ buckets

Atleast 782 buckets required
Since directory is always power of 2, so it will have atleast $2^{10} = 1024$ entries.

So lowest possible global Depth = 10

(b) Average occupancy.

If local depth of all buckets is same as global depth, then each directory entry points to unique bucket so total buckets = $2^{10} = 1024$

They will have capacity = 1024 x Bfr
= 1024 x 128
Total capacity = 131072

Avg occupancy of bucket = $\frac{100,000}{131072} \times 100 = 76.29\%$