Declaration and statement of authorship

I, bearing Roll No. 106119100, agree and acknowledge that:

1) The assesment was answered by me as per instruction applicable to each assessment, and that I have not resorted to any unfair mean. to deliberately improve my performance

2) I have neither impersonated anyone, nor have been impersonated by any person for the purpose of assessment.

Signature : Rpandey

Full Name : Rajneesh Pandey

Roll No : 106119100

Sub code : CSPC52

Mobile No : 8290968008

03/12/2021

ENDSEM EXAM

CSPC52 - DBMS

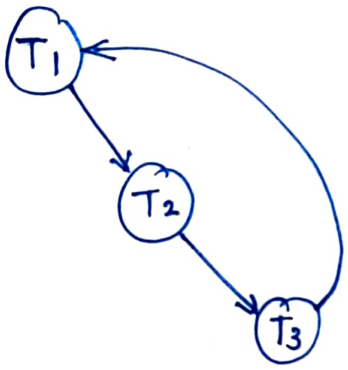
106119100

Rajneesh Pandey

Question (1)

(a)

precedence graph,



we can see here clearly that a cycle exists, hence, it is not conflict serializable.

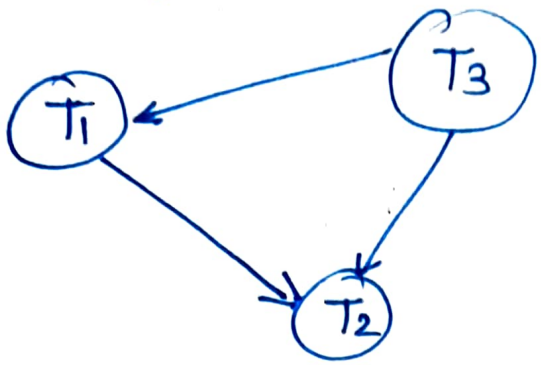
(i) Initial Read → as only one read is happenning, this satisfies

(ii) Final Write → as only a single write is happening, this satisfies.

(iii) update Read → This fails, because in any serial schedule, due to the cycle, there would be update conclict.

Hence, it is NOT view serializable.

(b) precedence graph:



we can clearly see that there is no cycle here.

Hence, it is conflict serializable.

Topological sort:

T_3, T_1, T_2

so, the serial schedule would be

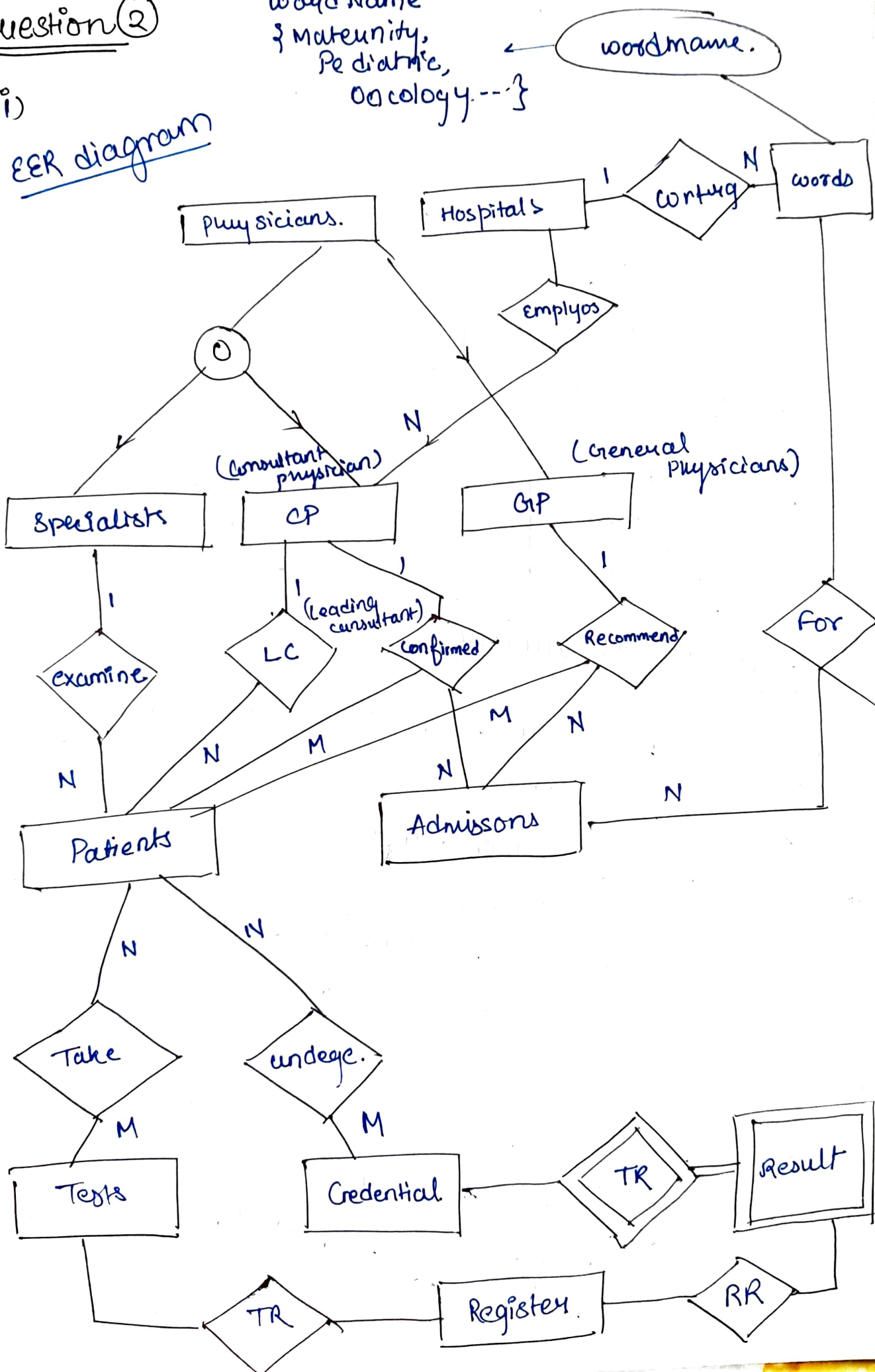
T_3, T_1, T_2 Ans

As, it is conflic serializable, hence it is also view serializable.

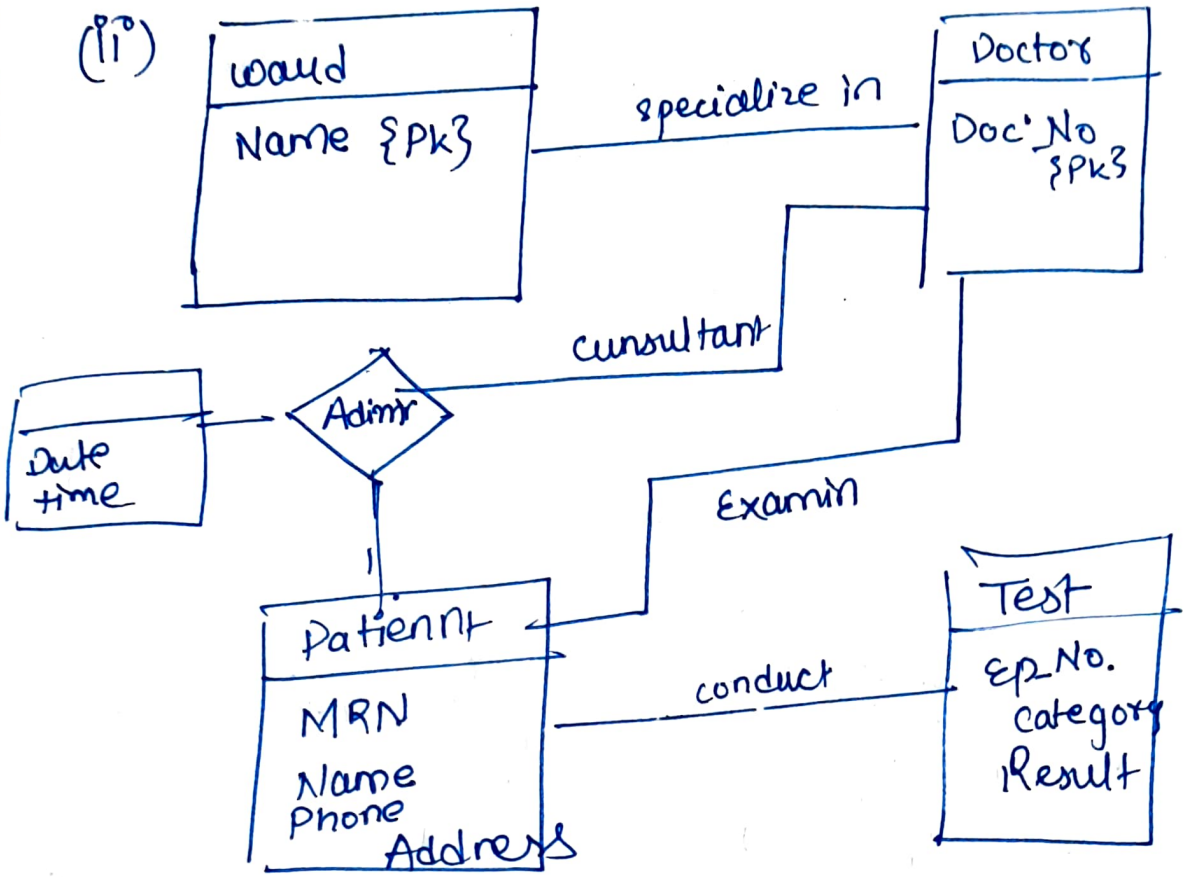
Question ②

(i)

EER diagram



(ii)



(iii) four sql query:

(a) SELECT d.name as "Department",
p.name as "physician"
from department d,
physician p.
where d.head = p.employeeid;

physicians who are the head of department

(b) Nurses not Registered

SELECT * FROM nurse
where registerd = 'false';

(c) Patient book appointment with at least with one doctor

```
SELECT count(DISTINCT patient) AS
        "No. of patients taken at least
                                one apointmen"
FROM appointment;
```

(d) count No. of room available

```
SELECT count(*) "Number of available
                        rooms"
    FROM room
    where unavailable = 'false';
```

(e) More SQL queries

```
SELECT * FROM Patient
SELECT * FROM Doctors
SELECT * FROM ward
```

Question (3)

$R \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\}$

FD: $\{$
$A_1A_2 \longrightarrow A_3$
$A_1A_3 \longrightarrow A_2$
$A_1A_4 \longrightarrow A_3$
$A_2 \longrightarrow A_4$
$A_2A_3 \longrightarrow A_1$
$A_5 \longrightarrow A_7$ $\}$

Now, if we find the candidate keys.

$$\begin{cases} A_1A_2A_6A_8^+ = A_1A_2A_3A_4A_5A_6A_7A_8 & -(1) \\ A_1A_3A_6A_8^+ = A_1A_2A_3A_4A_5A_6A_7A_8 & -(11) \\ A_2A_3A_6A_8^+ = A_1A_2A_3A_4A_5A_6A_7A_8 & -(111) \end{cases}$$

Candidate key

Now, to decompose into, BCNF using the rule:
Non-trivial $\alpha \longrightarrow \beta$, decompose as

(i) $\alpha \cup \beta$
(ii) $R - (\beta - \alpha)$

→ For every non trivial FD that's not superkey, decomposes.

(i) $A_1A_2 \longrightarrow A_3$

Decompose R as $R_0(A_1A_2A_3A_4A_5A_7)$
AND
$R_1(A_1A_2A_6A_8)$

(ii) $A_1A_3 \longrightarrow A_2$
A_1A_3 is superkey in R_1, no chance.

(iii) $A_1A_4 \longrightarrow A_5$

Decompose R_1 as $R_2(A_1, A_4, A_5A_7)$
AND
$R_3(A_1A_2A_3A_4)$

(iv) $A_2 \longrightarrow A_4$.

R_3 is split as $R_4(A_2, A_4)$ and
$R_5(A_1, A_2, A_3)$

(v) $A_2A_3 \longrightarrow A_1$
this is already a superkey in R_5

(vi) $A_5 \longrightarrow A_7$
Violates R_2 as A_5 is not a superkey
R_2 is split as $R_6(A_5, A_7)$ and
$R_7(A_1A_4A_5)$

So,

we get,

R_0 (A_1 A_2 A_6 A_8)
R_4 (A_2 A_4)
R_5 (A_1 A_2 A_3)
R_6 (A_5 A_7)
R_7 (A_1 A_4 A_5)

} BCNF is satisfied

Question (4)

F { $A_1 \longrightarrow A_2$, $A_2 \longrightarrow A_3$, $A_1 \longrightarrow A_3$, $A_1 A_2 \longrightarrow A_3$

so, we get FD: functional dependencies.

step 1: FD with only attribute as RHS

$A_1 \longrightarrow A_2$
$A_2 \longrightarrow A_3$
$A_1 \longrightarrow A_3$
$A_1 A_2 \longrightarrow A_1$

step 2: Remove Trivial FD's

$A_1 \longrightarrow A_2$
$A_2 \longrightarrow A_3$
$A_1 \longrightarrow A_3$

step 3: Minimizing the LHS, we obtained some as above

$A_1 \longrightarrow A_2$
$A_2 \longrightarrow A_3$
{ $A_1 \longrightarrow A_3$ } → transition

step 4: Removing Redundant FD's

$A_1 \longrightarrow A_2$
$A_2 \longrightarrow A_3$

→ minimal cover

we got the Answer

Question (5)

$F: \{A \longrightarrow B,\ BC \longrightarrow D\}$

Attributes $\{A, B, C, D\}$

so, for f^+, we get the following.

trivial:
- $A \longrightarrow A$
- $B \longrightarrow B$
- $C \longrightarrow C$
- $D \longrightarrow D$
- $E \longrightarrow E$

So, we now get the closures

$A \longrightarrow AB$, $(A)^+ : AB$.

$(B)^+ : B$

$(C)^+ : C$

$(D)^+ : D.$

$AB \longrightarrow AB$, $(AB)^+ : AB$

$AC \longrightarrow ABCD$ $(AC)^+ : ABCD$

$BC \longrightarrow BCD$ $(BC)^+ : BCD$

$AD \longrightarrow ADB$ $(AD)^+ : ADB$

$BD \longrightarrow BD$ $(BD)^+ : BD.$

$CD \longrightarrow CD$ $(CD)^+ : CD$

$ABC \longrightarrow ABCD$ $(ABC)^+ : ABCD$

$ABD \longrightarrow ABD$ $(ABD)^+ : ABD$

BCD ⟶ BCD : $(BCD)^+$: BCD

ACD ⟶ ABCD $(ACD)^+$: (ABCD)

ABCD ⟶ ABCD. $(ABCD)^+$: ABCD.

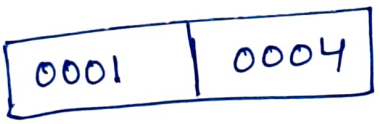
Hence, we obtaine A^+ as the union of all the closures of the attributes.

Question (6)

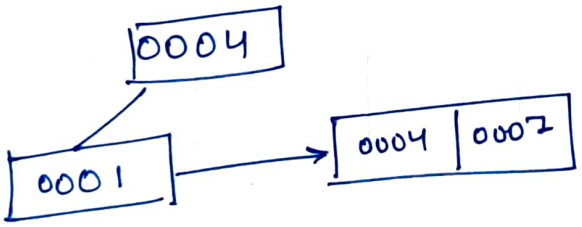
Insertions in B^+ tree.

i) 1

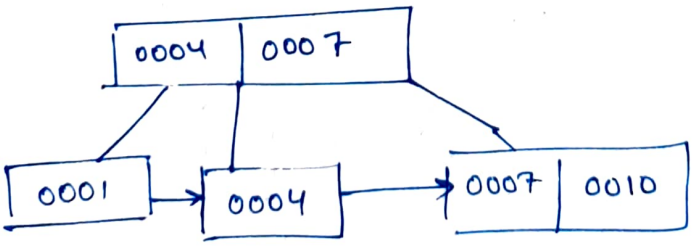
(ii) 4



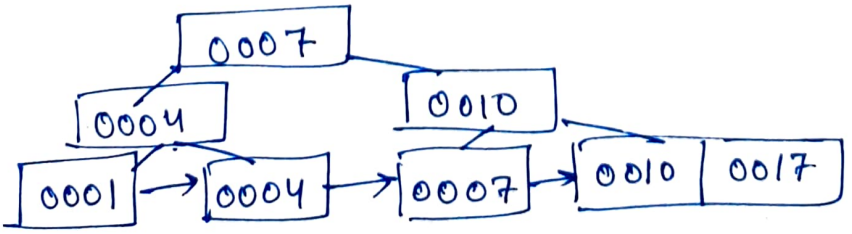
(iii) 7



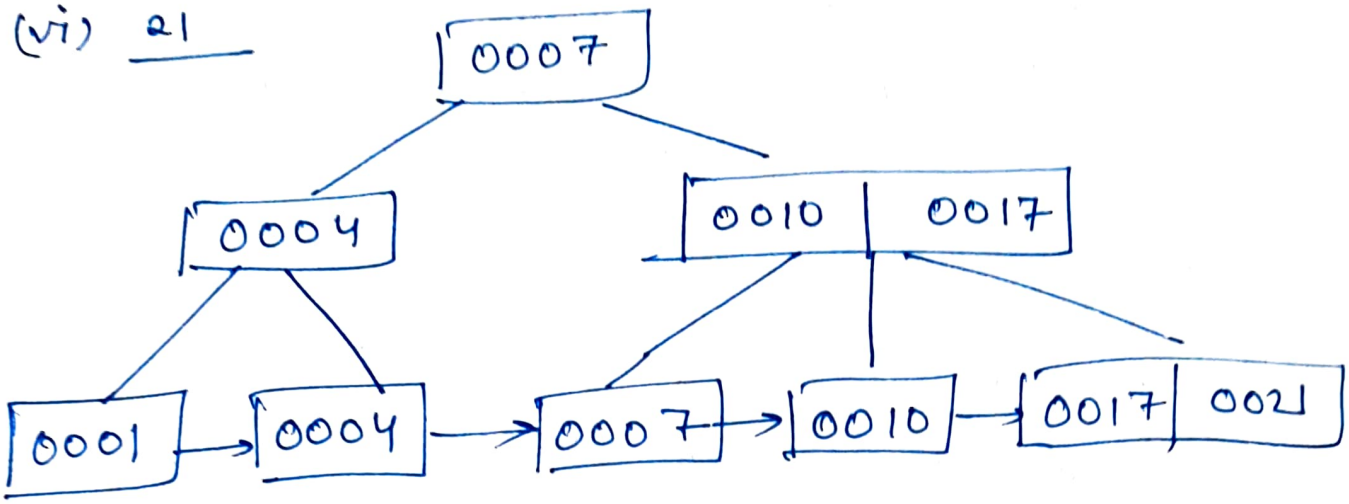
(iv) 10



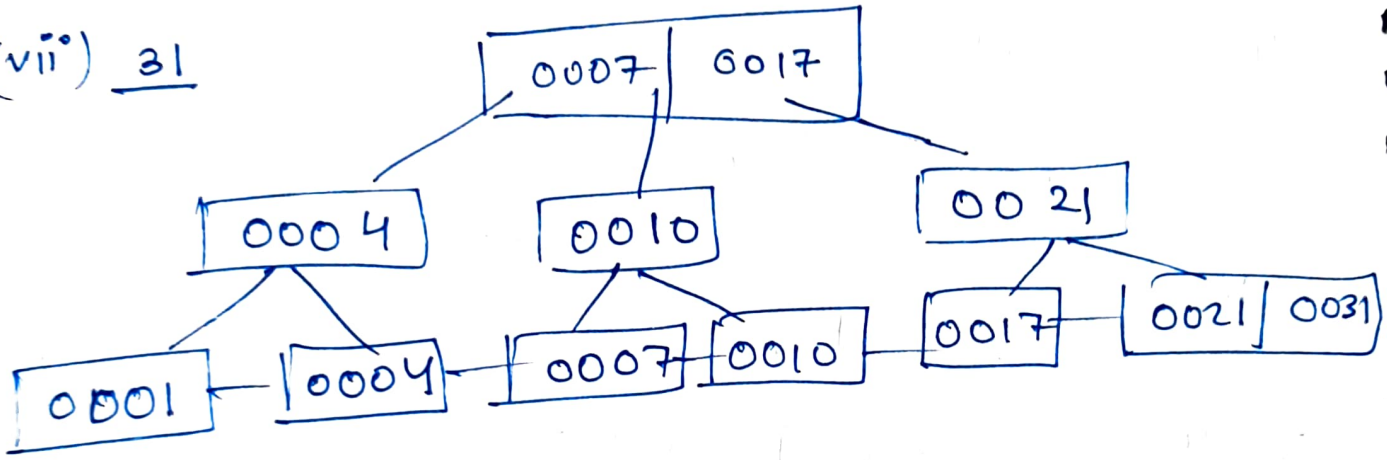
(v) 17



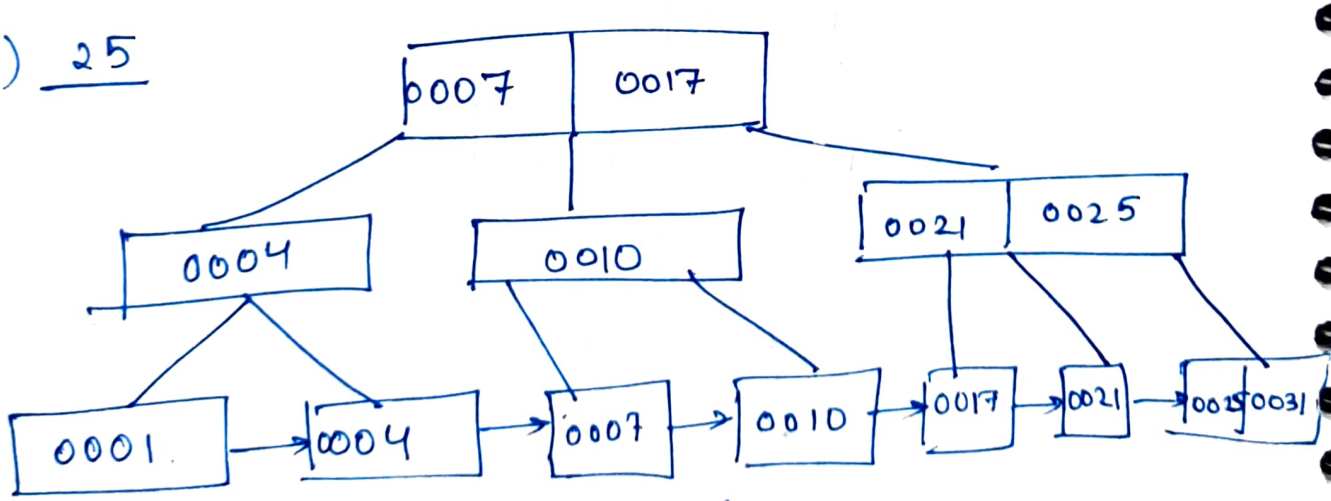
(vi) 21



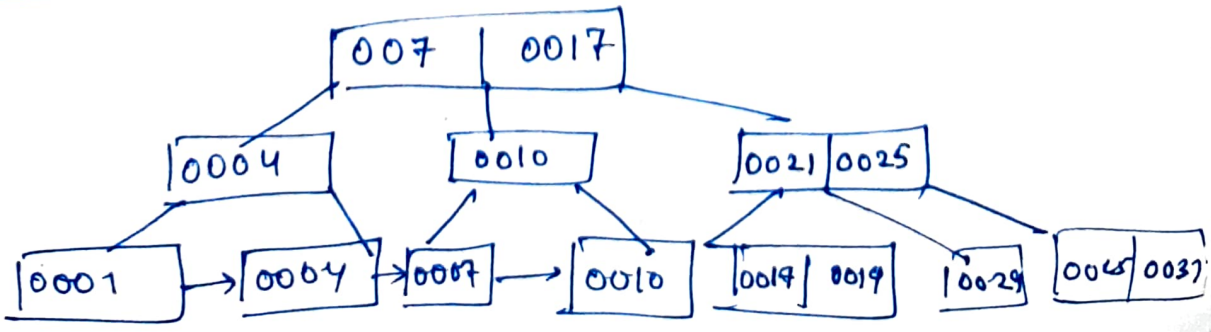
(vii) 31



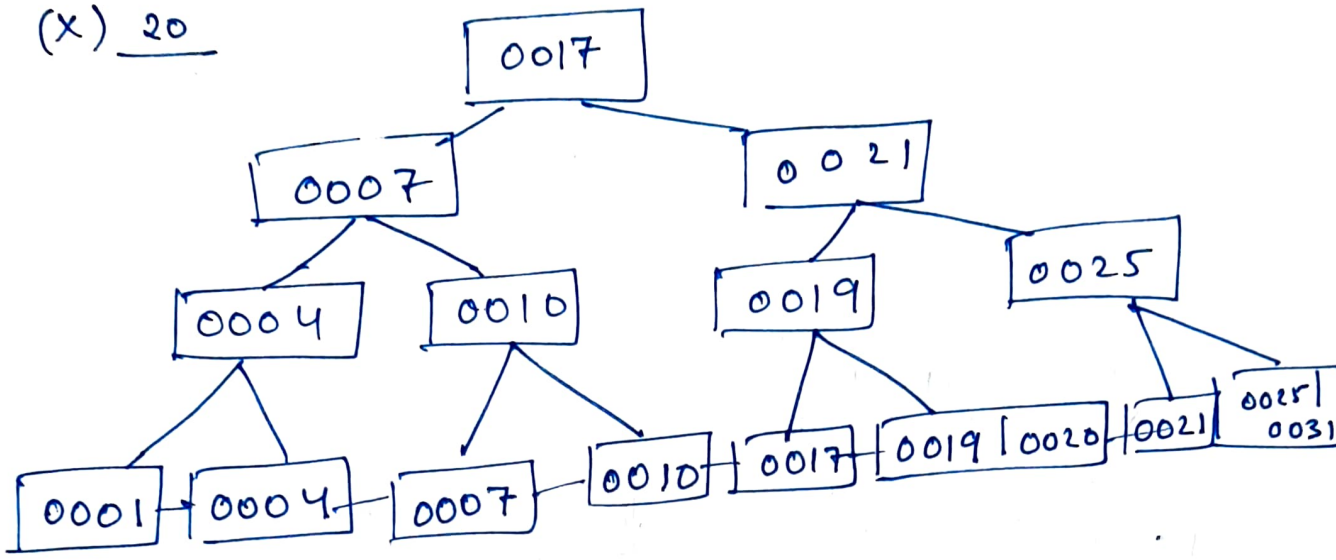
(viii) 25



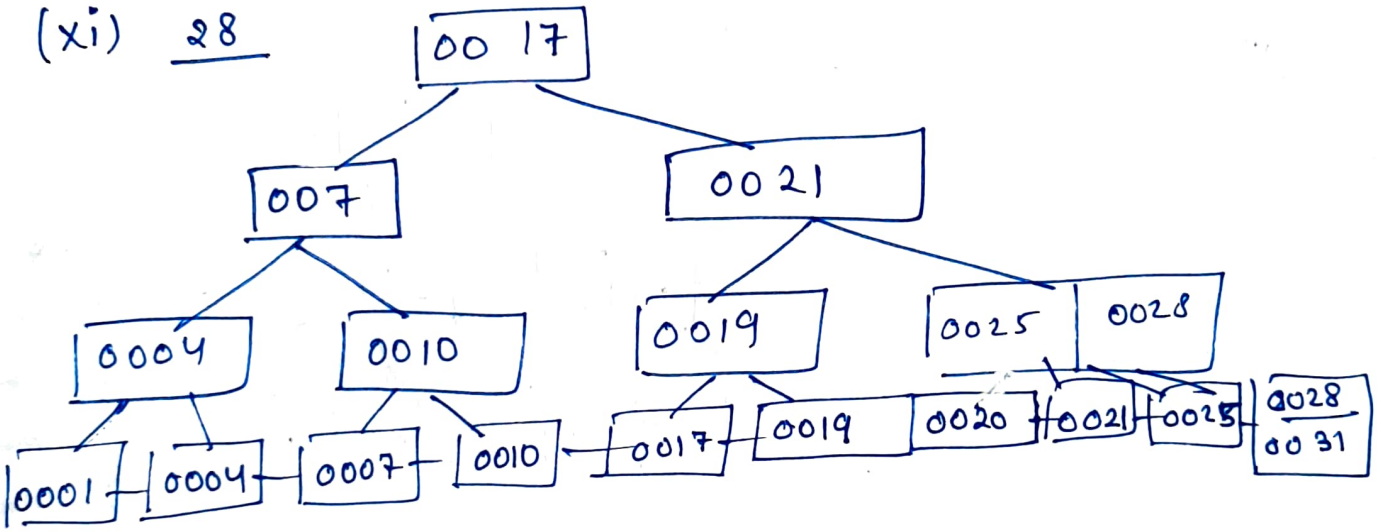
(ix) 19



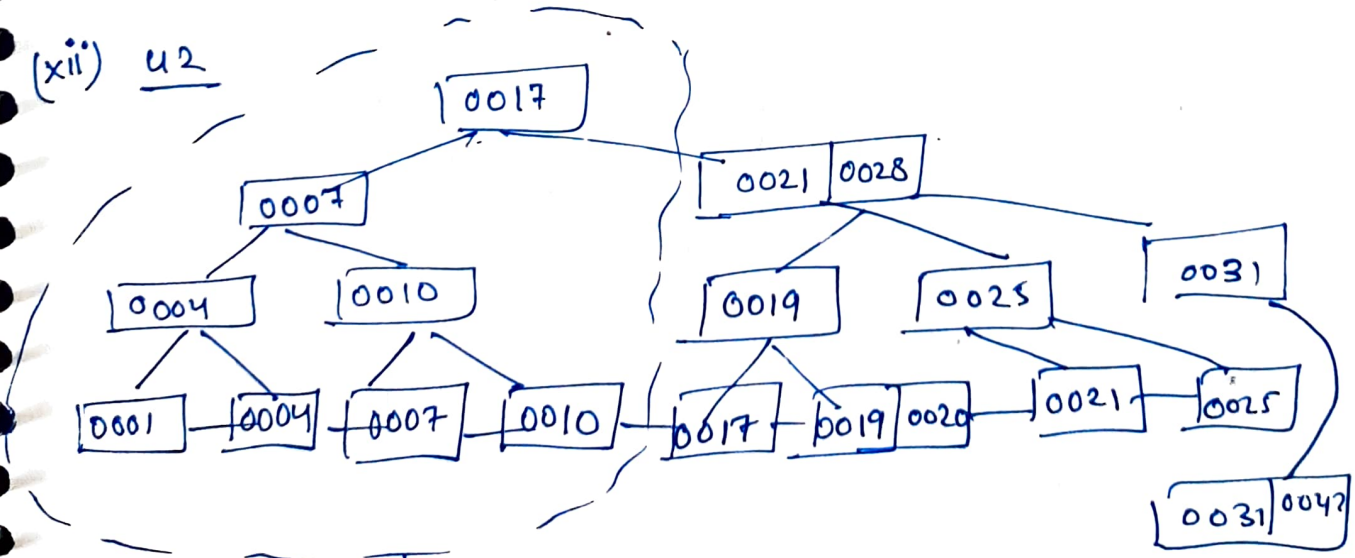
(x) 20



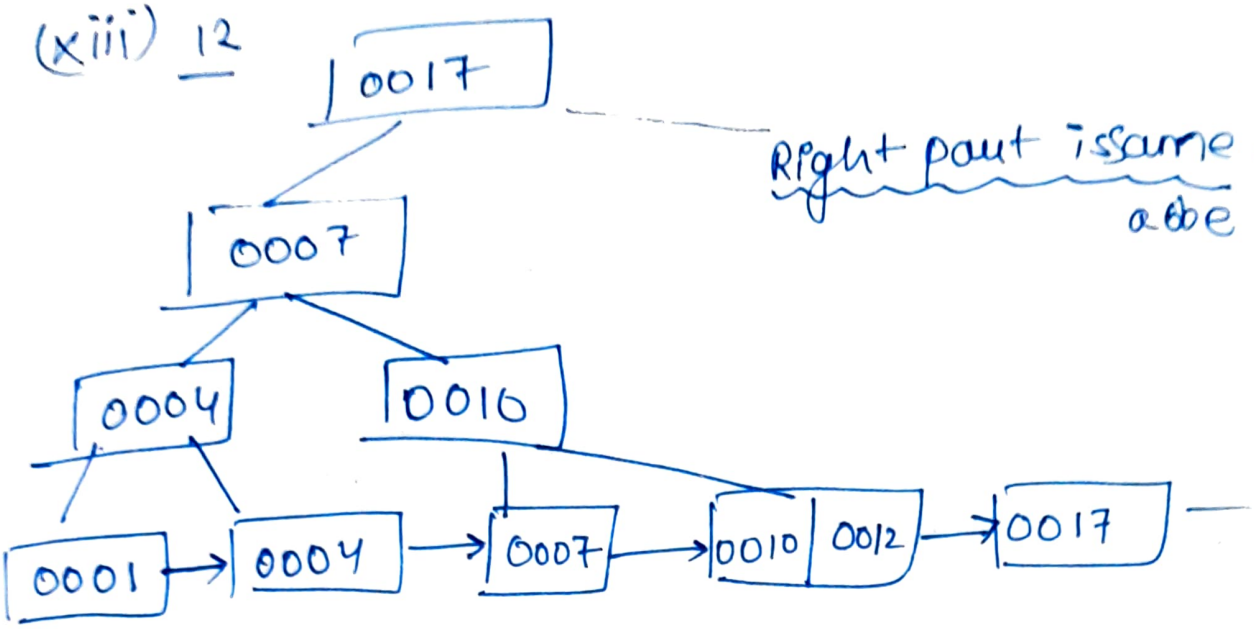
(xi) 28



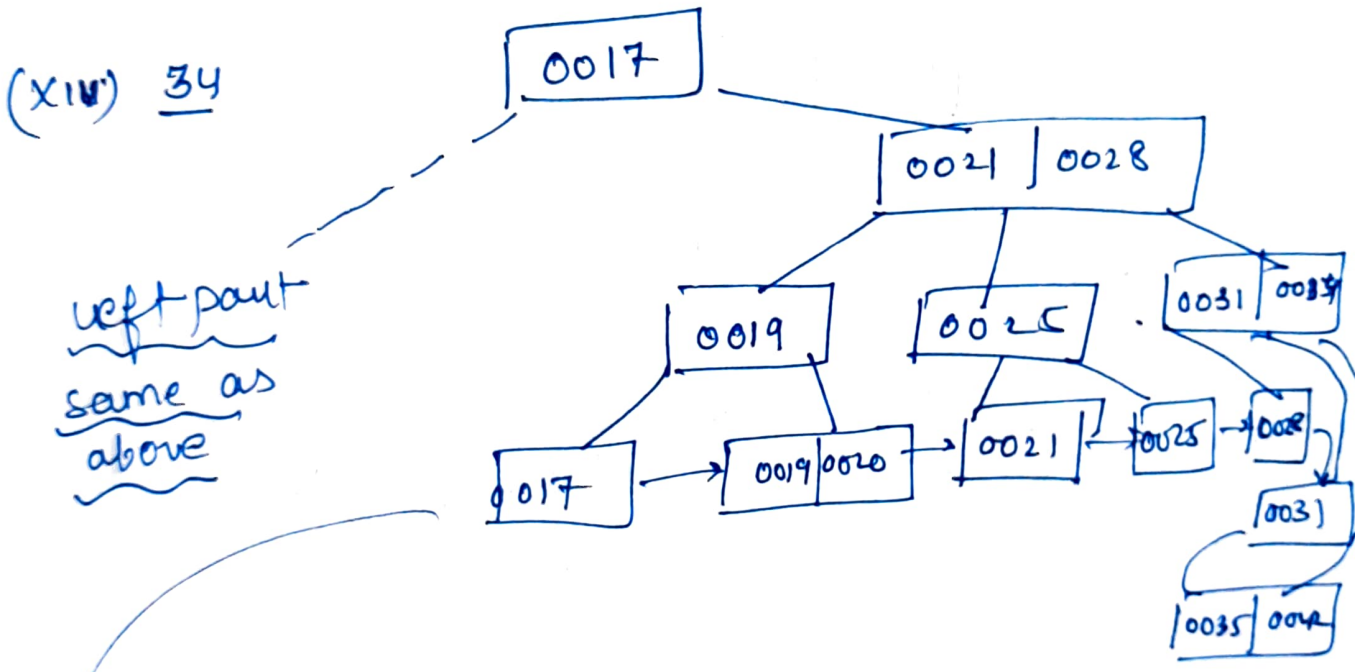
(xii) 42



(xiii) 12



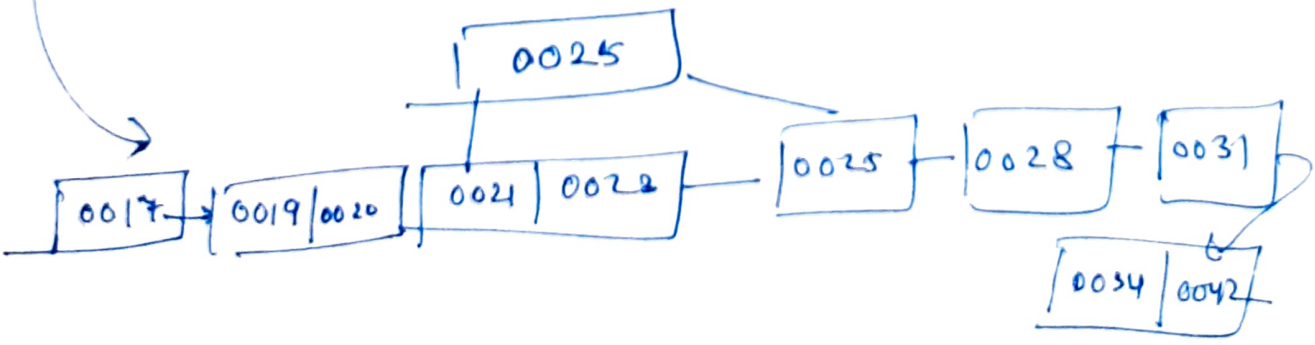
(xiv) 34



(xv) 23

same as above

inserting at the last leaf node

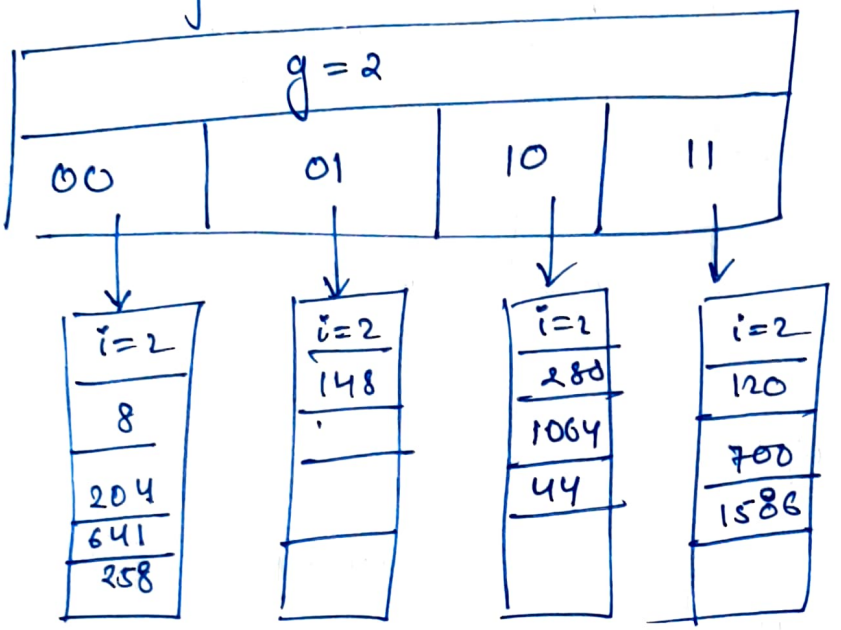


Hence all insertion done.

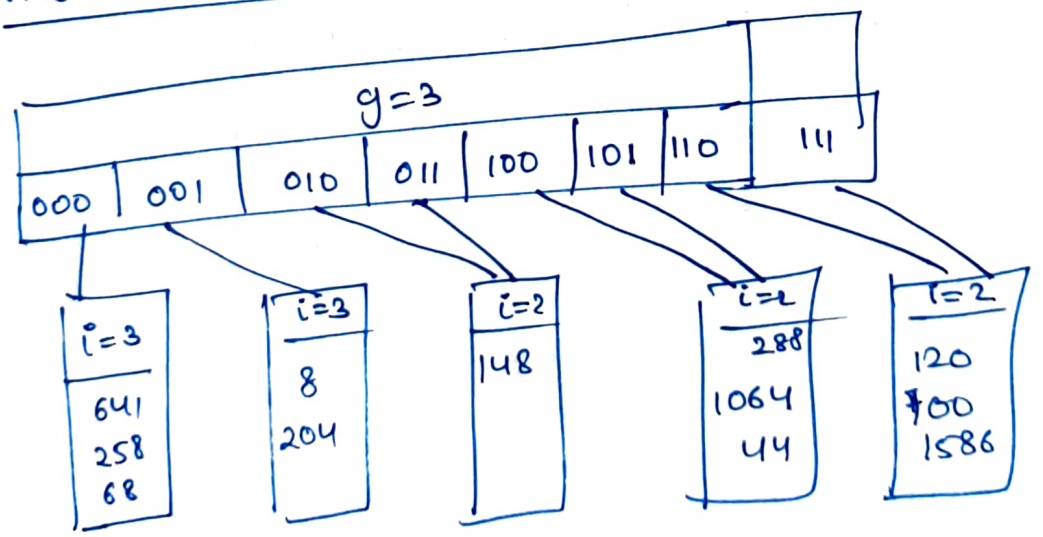
Question (7)

6-bit hash $\longrightarrow 2^6 = 64$.

→ Directory and Bucket structure.

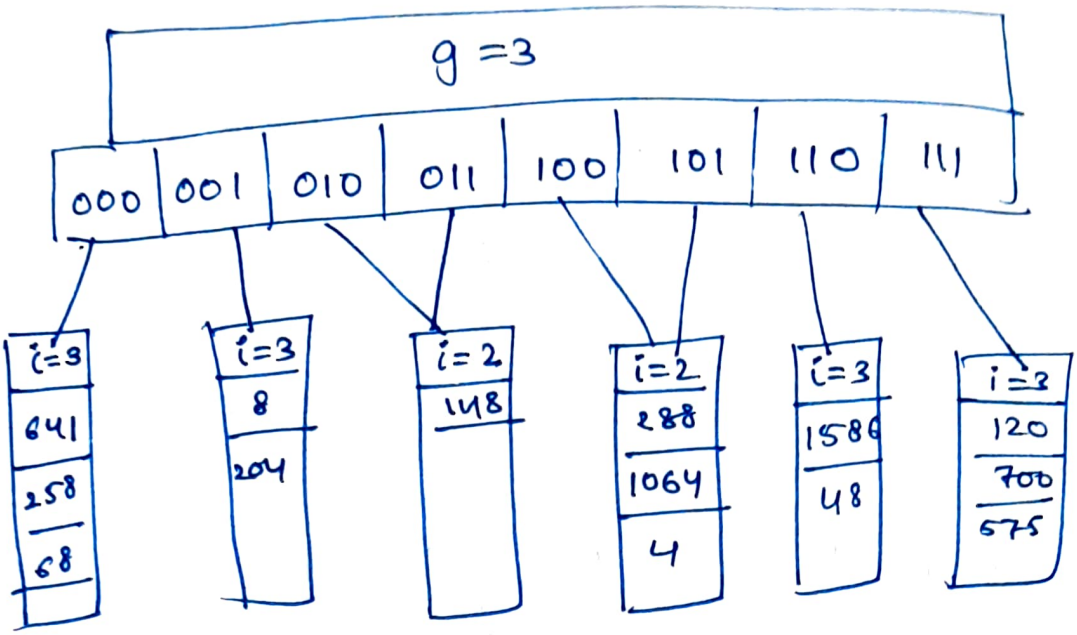


(i) insert 68 : 68 mod 64 = 4

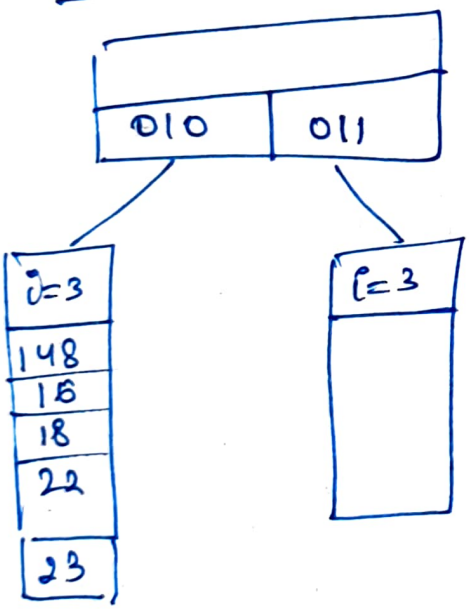


Bucket and directory split

(ii) inserting 48 and 575 → Bucket split
no directs split.



(iii) Insert 16, 18, 22, 23



multiple splits

16 mod 64 = 010000

18 mod 64 = 18

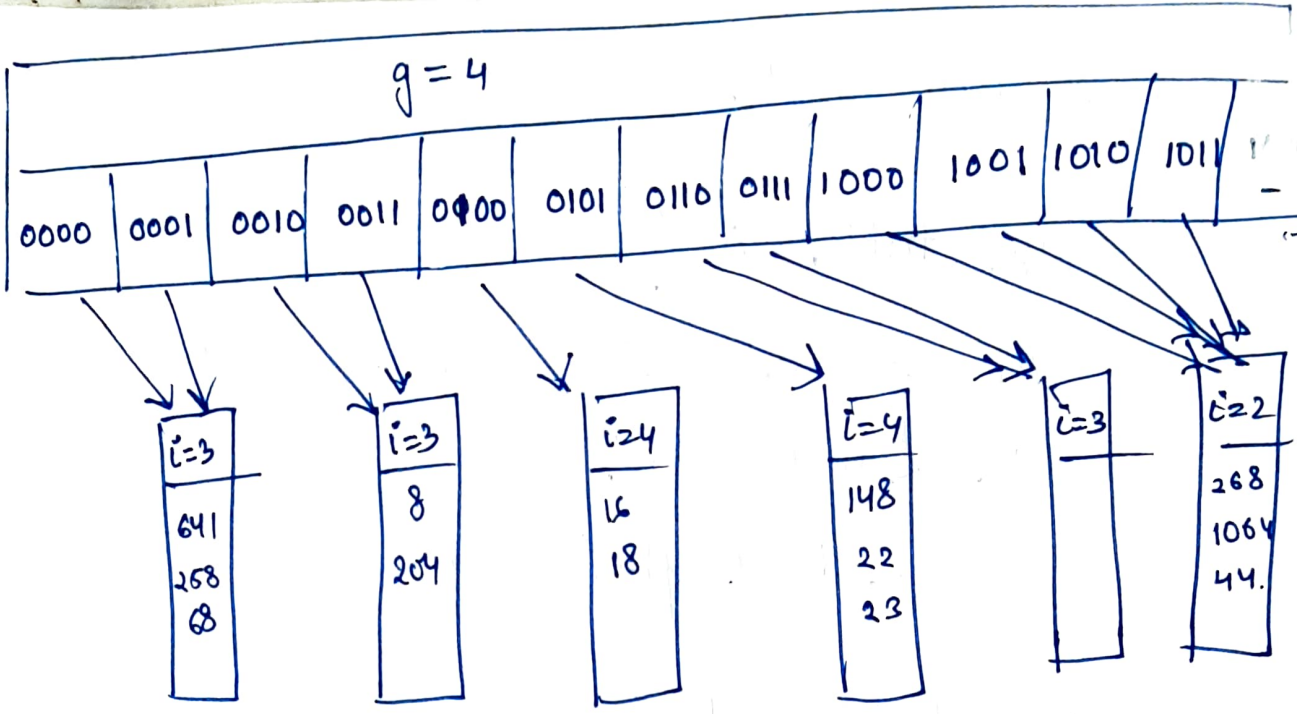
22 mod 64 = 22

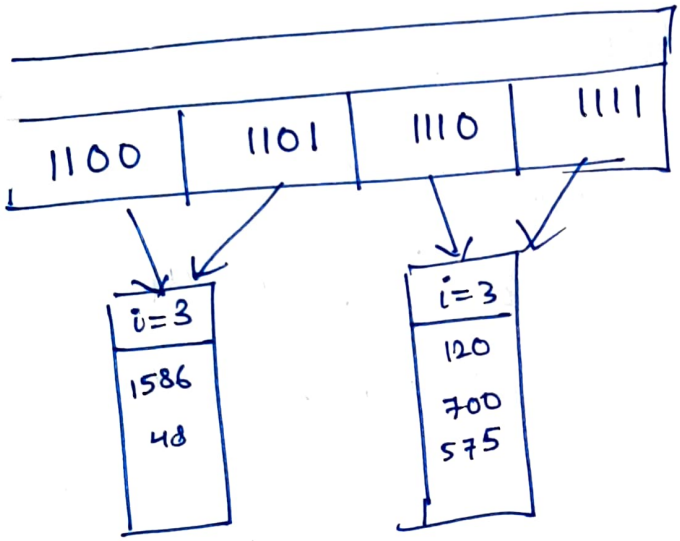
23 mod 64 = 23

setting i = 3 gives the table as

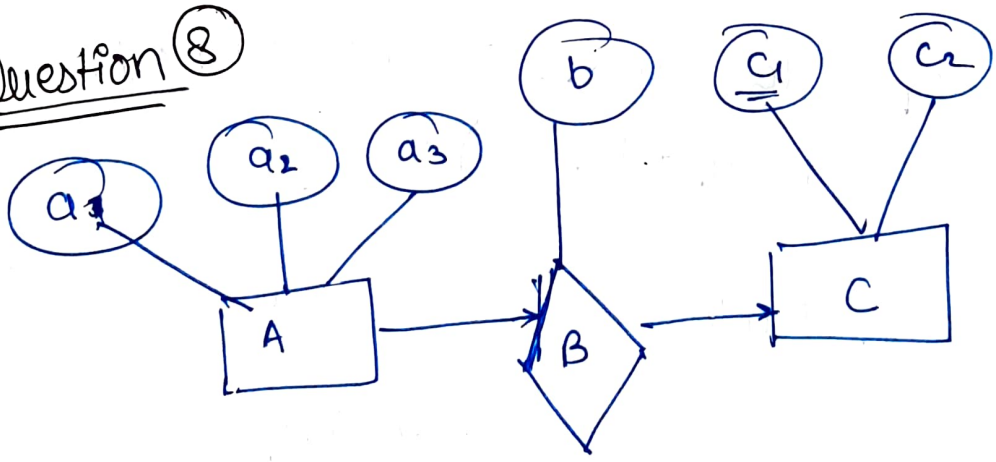
(iv) multiple splits

setting i=4 & g=4





Question (8)



many to one mapping from A to C via B.

→ Maximum tables = 3 :

```
create table A (
    a1  <datatype>  Primary key,
    a2  <datatype>,
    a3  <datatype> );

create table B (
    a1  <datatype>
    a1  <datatype>
    b   <datatype>
    foreign key a1 refrence A(a1),
    foreign key c1 reference c(c1));

create table C (
    c1 <datatype> primary key,
    c2 <datatype> );
```

→ Minimum tables = 2

we shift to the many side for the relationship.

```
create table A (
        a1 <data type>  primary key.
        a2 <datatype>
        a3 < data type>
        b  < data type>
        c1 < data type>
foreign key a1 references C(c1));

create table c (
        c1 int primary key,
        c2 int );
```

Question (9)

a) A B C D E

No tuple as C+D together do not match

(b)

A	B	C	D	D'	E
3	4	2	6	1	3
2	2	5	1	4	5
4	2	5	3	4	5
3	4	2	6	5	7

(c) $\Pi_{BE}(\Pi_{B,C}R) \bowtie (\sigma_{E=7}S)$

B	C
2	3
2	5
4	2
2	5

$\bowtie$

D	E
1	2 3
4	5 5

B	C	D	E
	5	4	5
2			
2	5	4	5
4	2	1	3

Hence, we get

B	E
2	5
2	5
4	3

ANS.

(d) $\Pi_{A,B}R - \rho_{S(A,B)}(\Pi_{D,C}S)$

we, get by performing the projection,

ans:

A	B
2	2
4	2

Question (10)

T_5 : already committe before the checkpoint, so no need to do anything.

T_6 : Not committed ∴ undo

T_7 : Committed before crash

∴ Redo

T_8 : Committed before checkpoint, no need to do anything.