Ans. to the Q. No- 01

$R(A, B, C, D, E)$

$A \rightarrow B, C, D$

$B \rightarrow E, D$

$D \rightarrow A, C$

ⓐ To find the candidate keys of the relation, we have to find the closure of each attribute -

$ABCDE^+ = \{A, B, C, D, E\}$

(sk) $A^+ = \{A, B, C, D, E\}$

(ck)

(sk) $B^+ = \{B, E, D, A, C\}$

(ck)

(sk) $D^+ = \{D, A, C, B, \cancel{D} E\}$

(ck)

$\therefore$ Candidate keys $= \{A\}, \{B\}, \{D\}$

ⓑ Step-1:

$A \rightarrow B$ ✓

$A \rightarrow C$ × redundant

$A \rightarrow D$ × redundant

$B \rightarrow E$ ✓

$B \rightarrow D$ ✓

$D \rightarrow A$ ✓

$D \rightarrow C$

Step-2: Remove redundant FD -

$A \rightarrow B$

$B \rightarrow E$

$B \rightarrow D$

$D \rightarrow A$

$D \rightarrow C$

(Canonical cover)

$A^+ = A, C, D,$

$A^+ = A, B, D, E, C$

$A^+ = A, B, E, D, C$

$B^+ = B, D, A, C,$

$B^+ = B, E$

$D^+ = D, C$

$D^+ = D, A.$

#ⓒ From canonical cover

A → B

B → DE

D → AC

~~The relation is 2NF. Because, by defalt it's 1NF. There is no partial dependency, so 2NF. But have transitive dependency, so not 3NF.~~

The relation is BCNF.

Because, by defalt it's 1NF.

There is no partial dependency, so 2NF.

No transitive dependency, so 3NF.

Only super key determin, so BCNF.

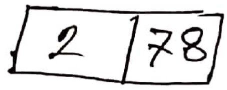
#ⓓ The relation is already in BCNF.

Ans. to the Q. No- 02

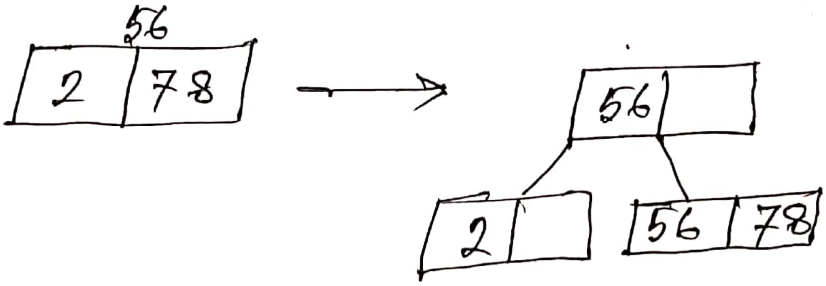
Data: 2, 78, 56, 1, 63, 12, 14, 18, 21, 8, 26, 10, 9, 13, 22, 46, 36

Order = m = 3

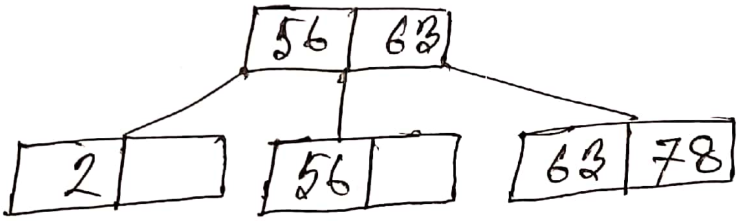
Insert: 2, 78



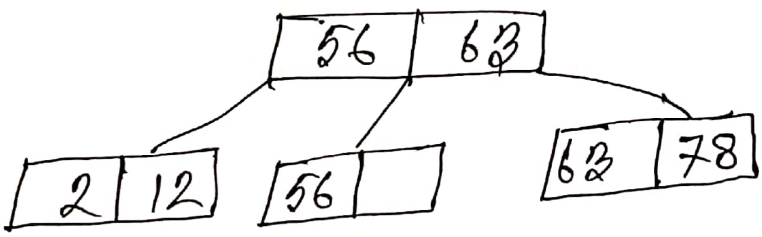
Insert: 56



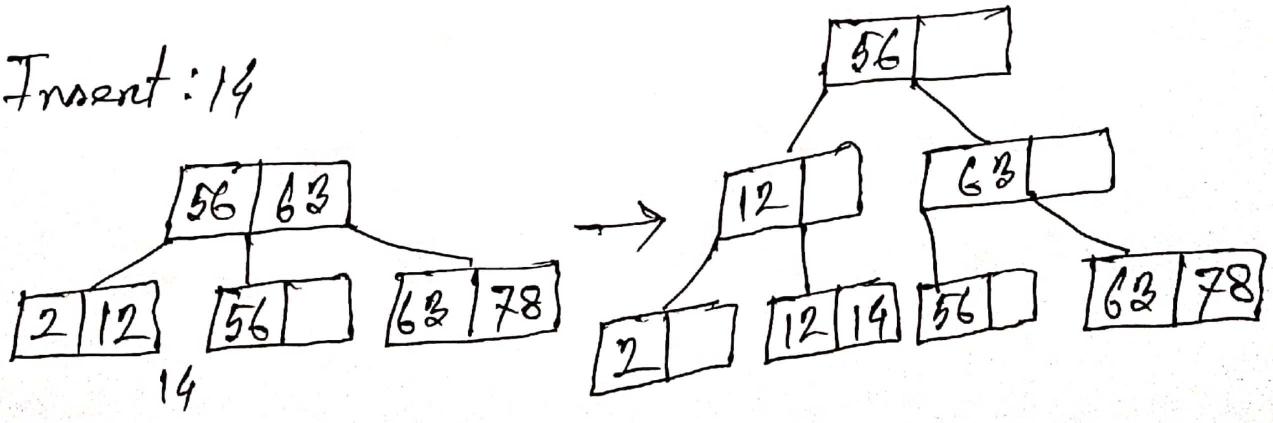
Insert: 63



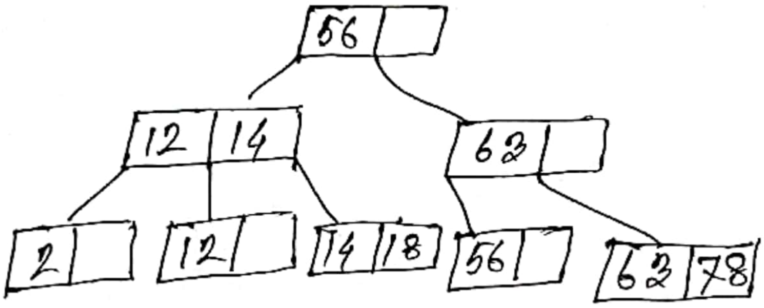
Insert: 12



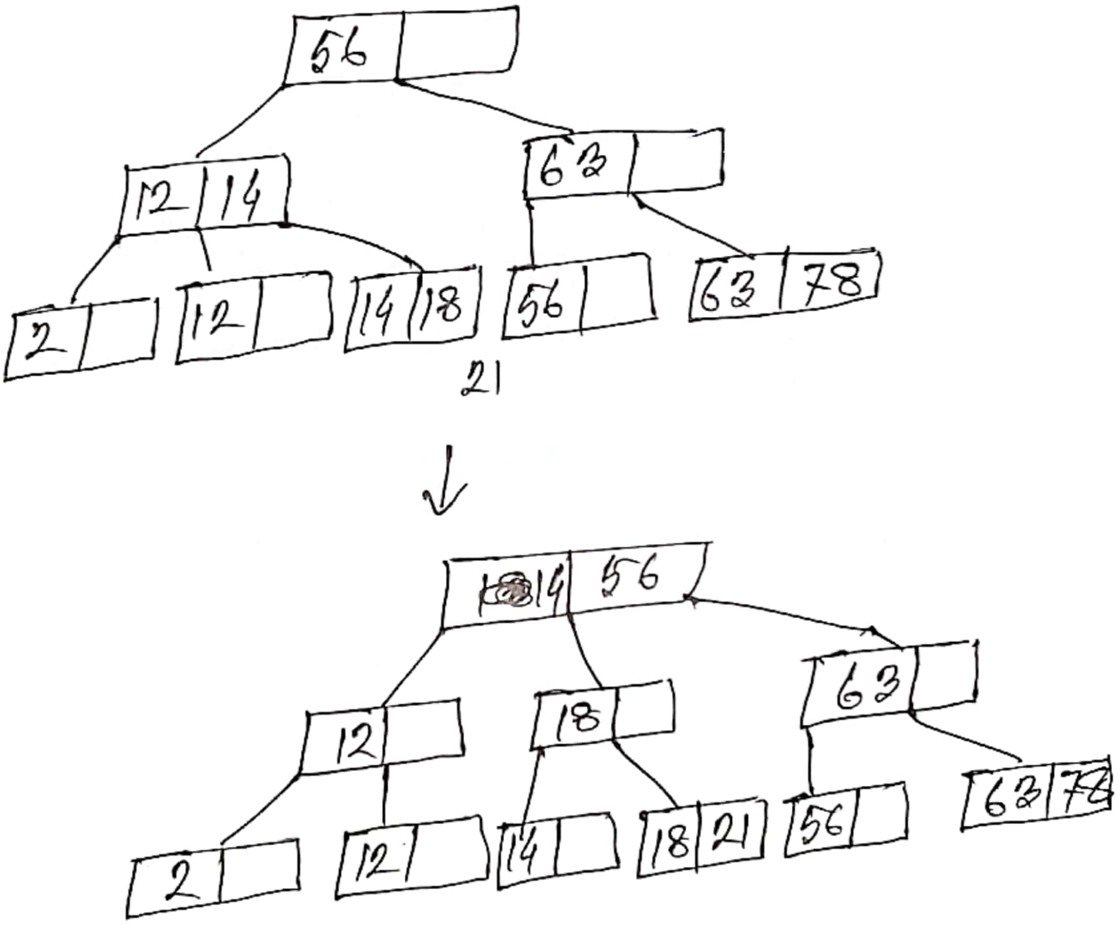
Insert: 14



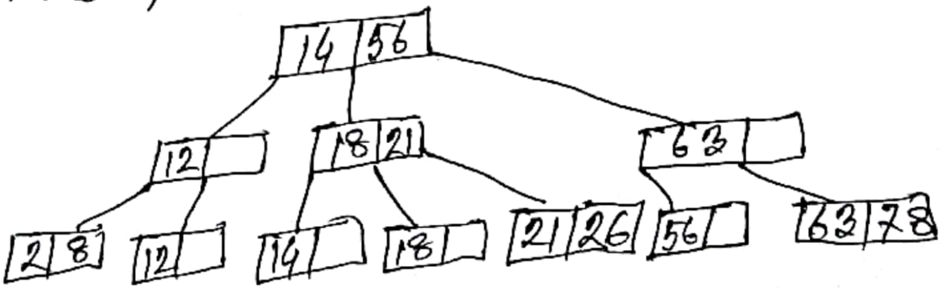
Insert: 18



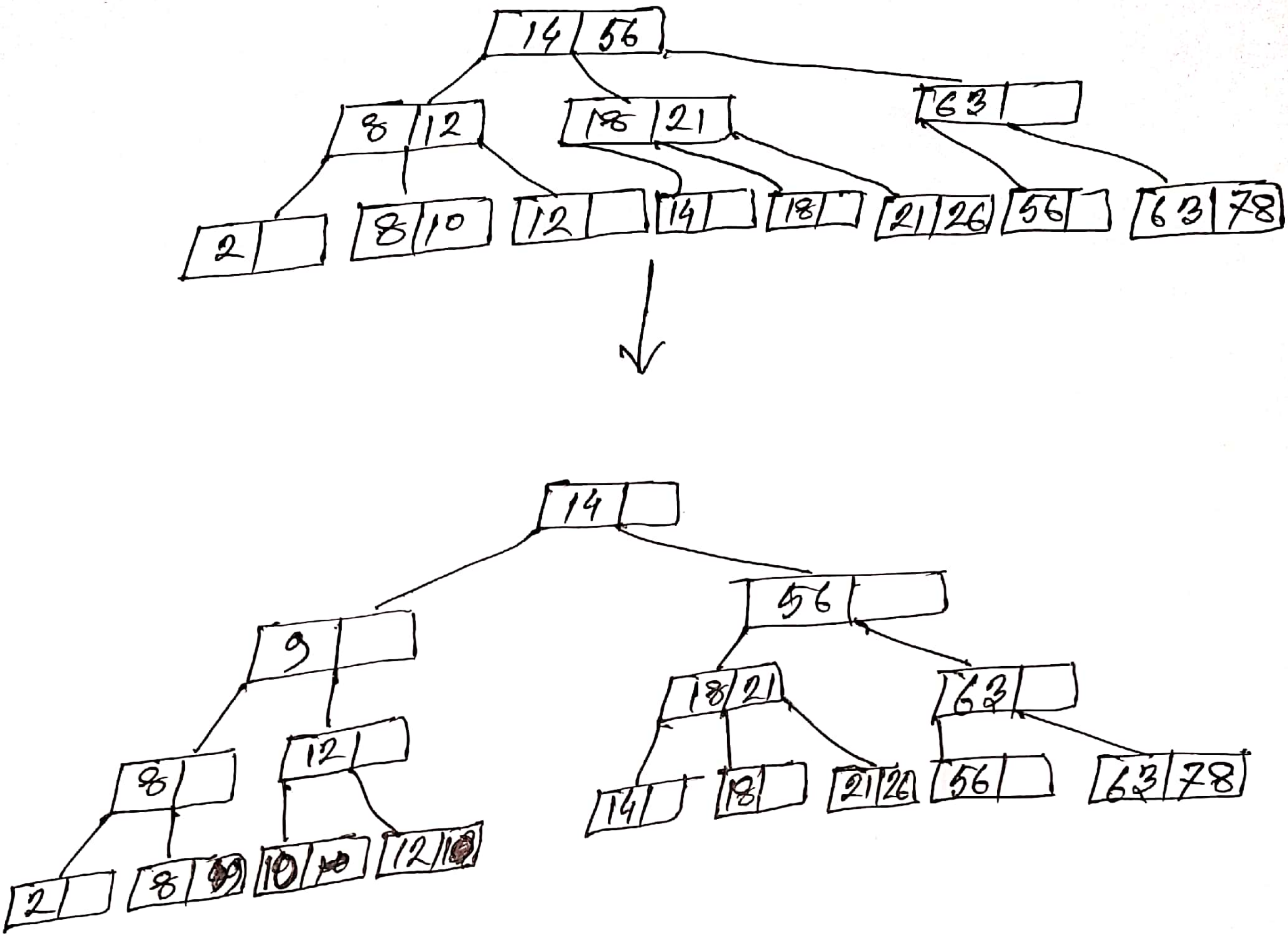
Insert: 21



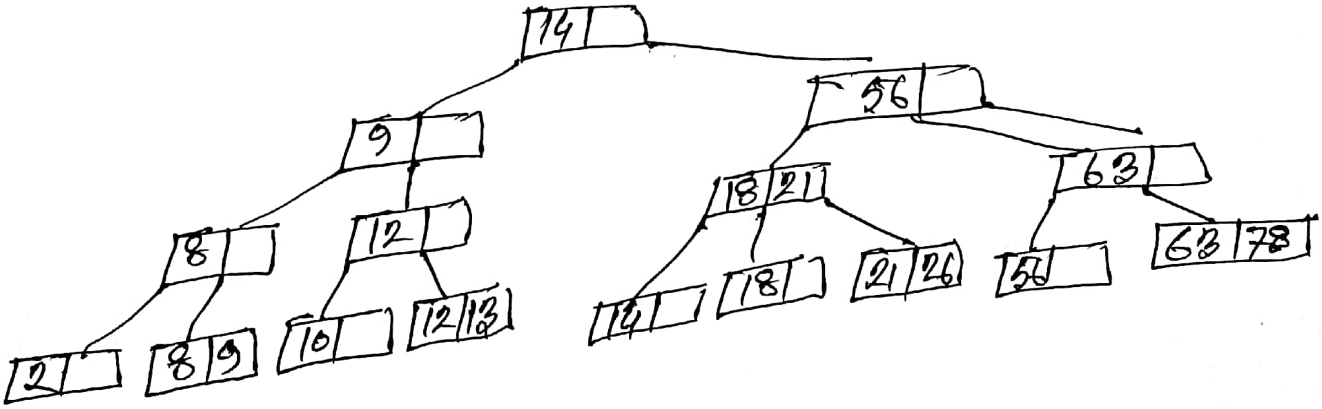
Insert: 8,26



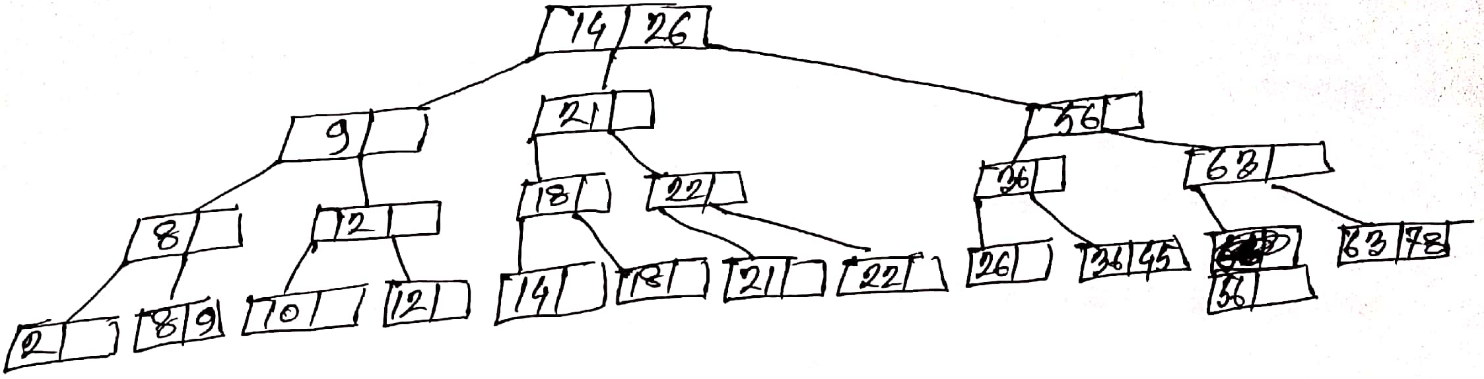
Insert: 10, 9



Insert: 13



Insert 22, 45, 36



Ans. to The Q. No-03 (Fall-22 (2.a))

Indexing helps in managing memory by reducing the amount of data that needs to be scanned during a search operation, which can result in more efficient use of memory.

Hash Function = key %7

5 %7 = 5
11 % 7 = 4
12 %7 = 5
19 %7 = 5
26 %7 = 5

28 %7 = 0
18 %7 = 4
33 % 7 = 5
24 %7 = 6
32% = 4

Index	Value
0	28 → NULL
1	
2	
3	24 → NULL
4	11 → 18 → 32 → Null
5	5 → 12 → 19 → 26 → 33 → → Null
6	24 → Null

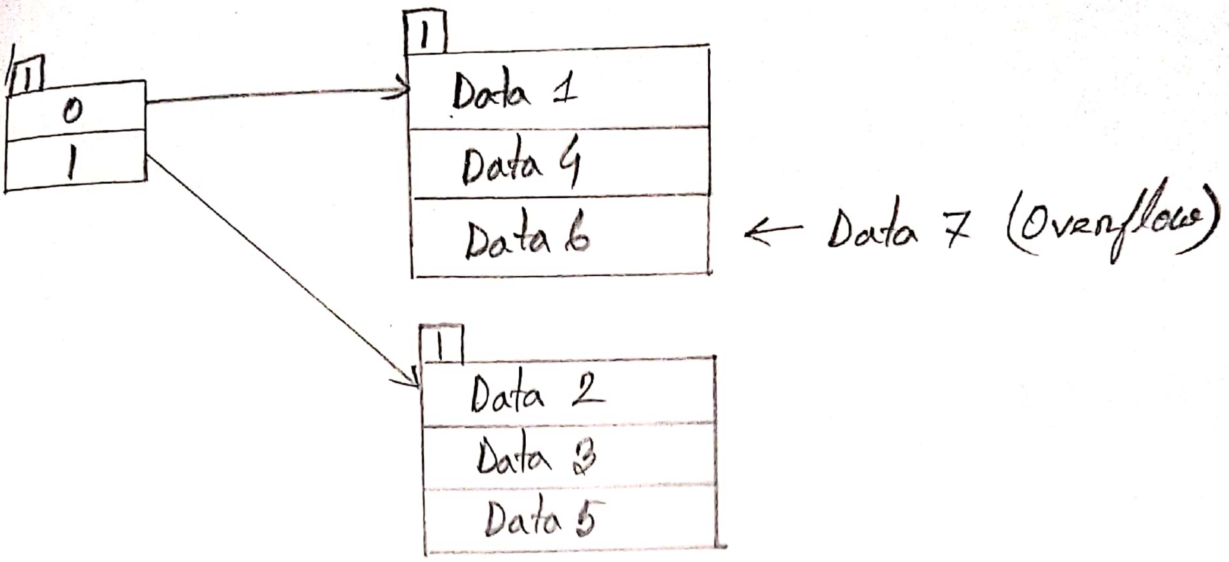
This hash function is not good for the given data. By using this hash function data could not be keep uniformly. Here some indexes makes long chain of data and some are null.

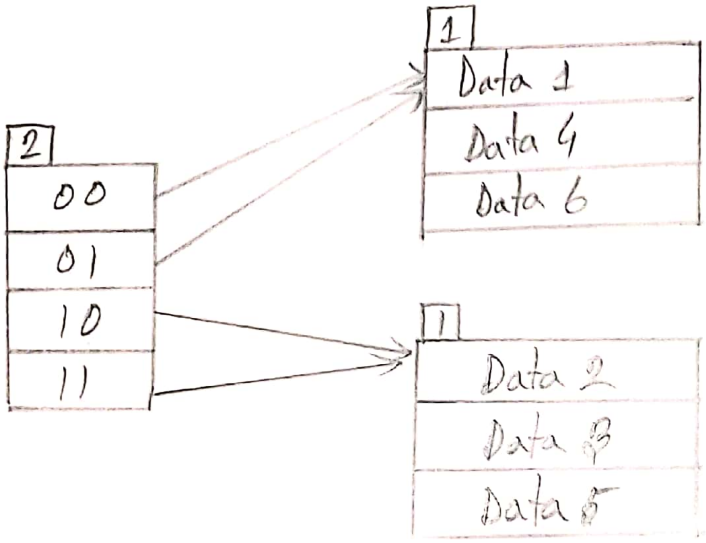
Ans. to the Q.No-03 (Fall-22(4))

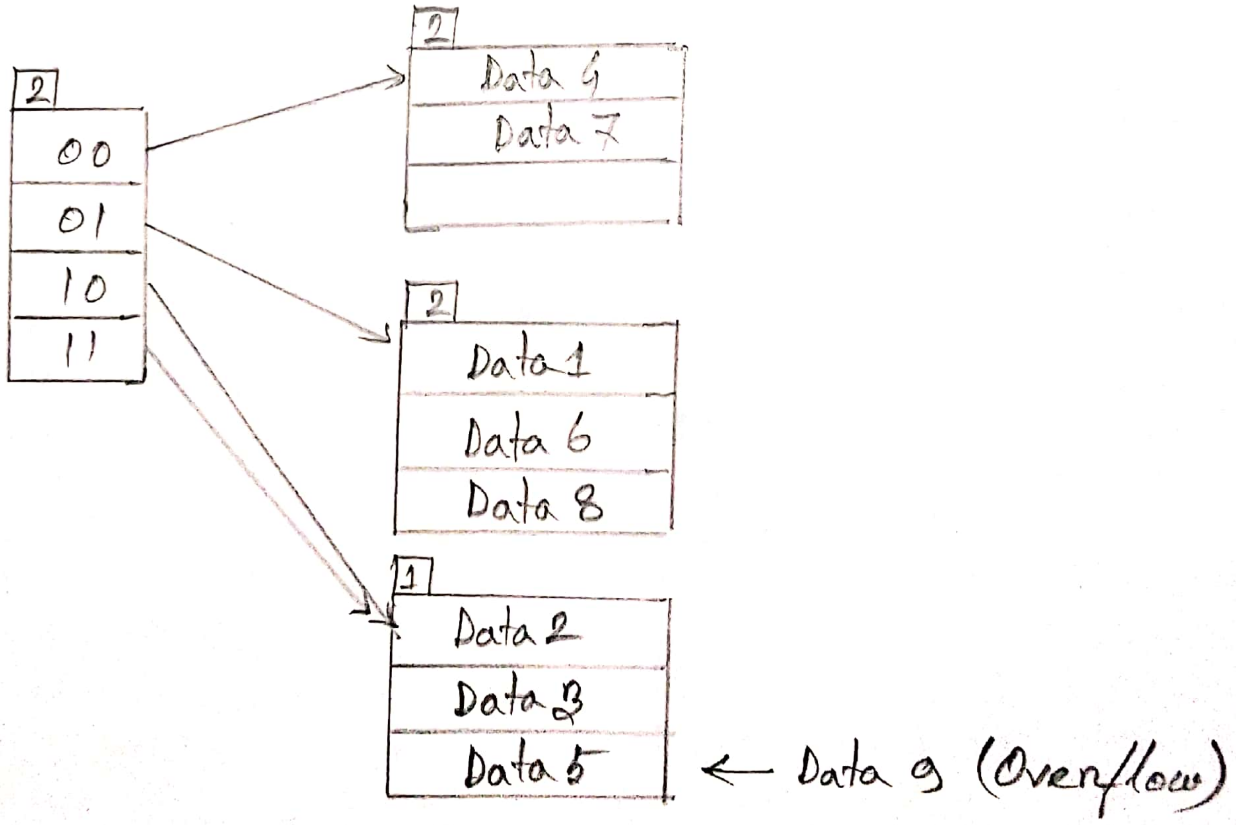
Data records	Search_key	Hash (Branch key)	Binary
Data 1	AFR	16	010000
Data 2	HDE	48	110000
Data 3	IYC	32	100000
Data 4	EFG	4	000100
Data 5	ADF	52	110100
Data 6	EFG	17	010001
Data 7	KNY	13	001101
Data 8	OKU	25	011001
Data 9	HMK	33	100001
Data 10	YGL	21	010101

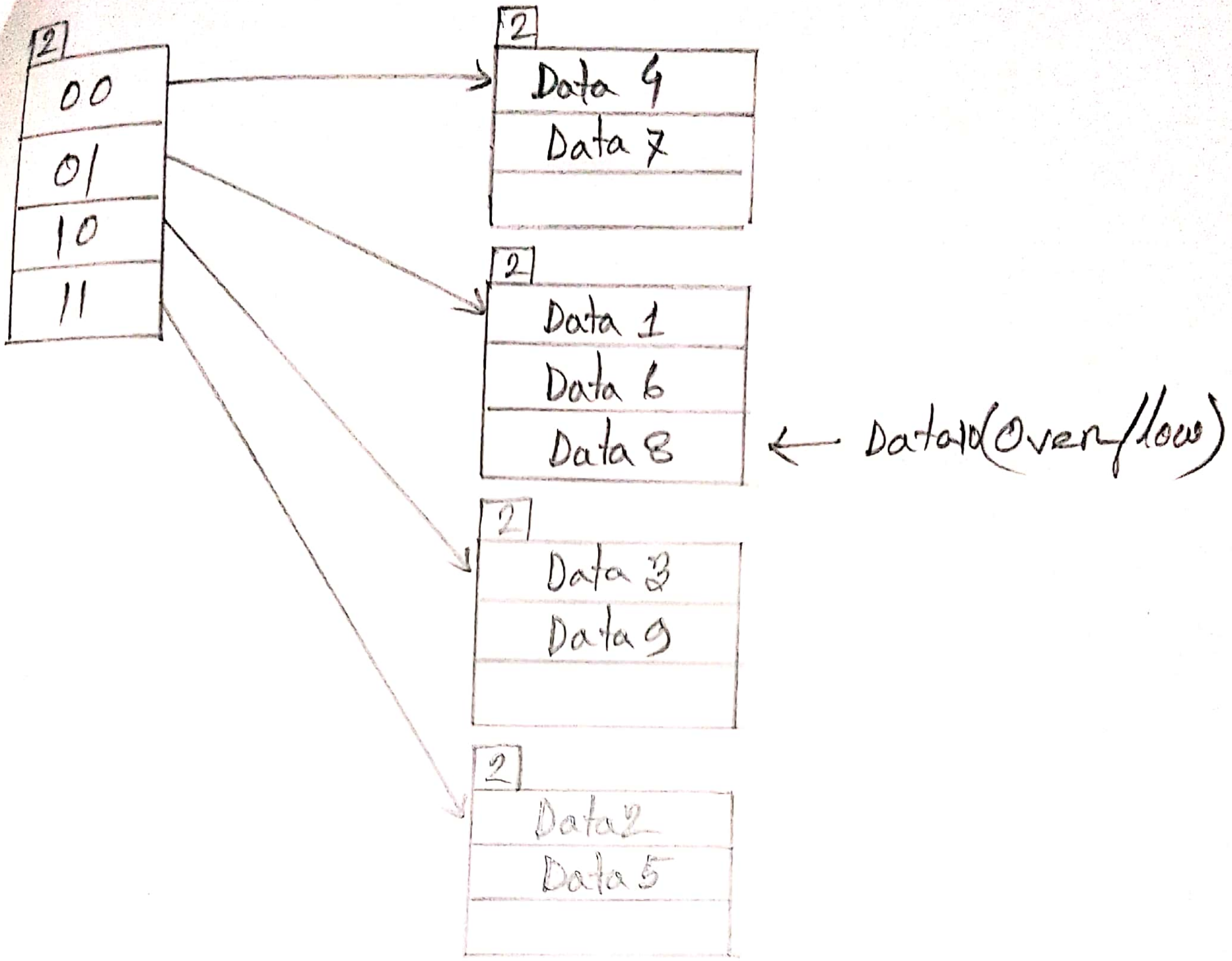
Here bucket size = 3

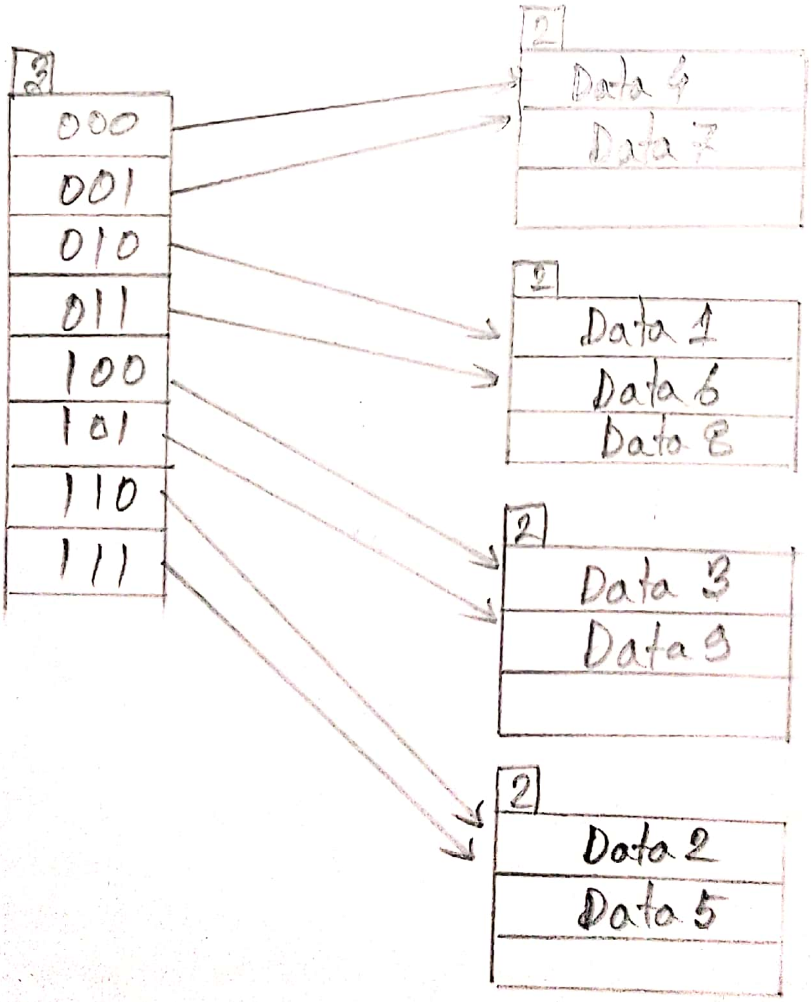
Initial global depth and local depth = 1

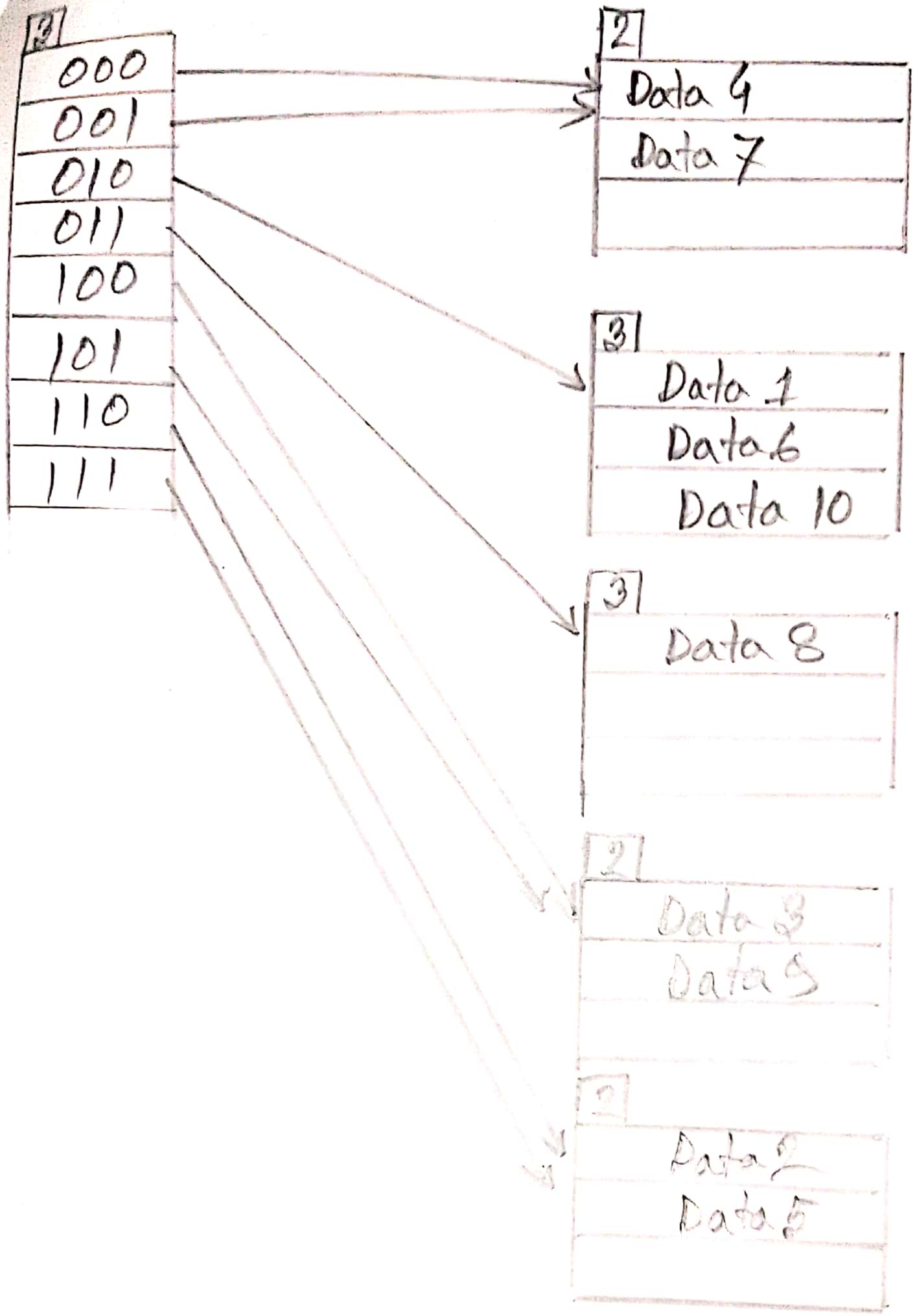

1
0
1
1
Data 1
Data 4
Data 6
← Data 7 (Overflow)
1
Data 2
Data 3
Data 5


2
00
01
10
11
1
Data 1
Data 4
Data 6
1
Data 2
Data 3
Data 5


2
00
01
10
11
2
Data 4
Data 7
2
Data 1
Data 6
Data 8
1
Data 2
Data 3
Data 5
← Data 9 (Overflow)


2
00
01
10
11
2
Data 4
Data 7
2
Data 1
Data 6
Data 8
Data10(Overflow)
2
Data 3
Data 9
2
Data2
Data 5


3
000
001
010
011
100
101
110
111
2
Data 4
Data 7
2
Data 1
Data 6
Data 8
2
Data 3
Data 9
2
Data 2
Data 5


3
000
001
010
011
100
101
110
111
2
Data 4
Data 7
3
Data 1
Data 6
Data 10
3
Data 8
2
Data 3
Data 9
2
Data 2
Data 5