Make-up Exam:→

PART-1:→

1.1 One record is composed of the following fields (size in bytes):→

(i) Record length: 4 bytes.
(ii) Birth date : 10 bytes.
(iii) SSN : 10 bytes
(iv) Patient ID: 10 bytes.
(v) Pointer to address: 4 bytes.
(vi) Pointer to patient history: 4 bytes.

No need for a pointer in case of name cause it always starts from the 35th byte in each record.

Assuming that no extra space was spent in any alignment work.

∴ Total = 42 bytes.

1.2 We will take the average of variable lengths

name → 30
Address → 50
history → 500

+42 from 1.1

30 + 50 + 500 + 42 = 622 bytes.

1.3 a.) :→

The repeating field is in the record itself :→

Record length : 4 bytes.
Pointer to address : 4 bytes.
Pointer to patient history : 4 bytes.
Birth Date : 10 bytes.
SSN : 10 bytes.
~~Pap~~ Patient ID : 10 bytes
name : variable length
address : variable length.
patient history : variable length.
cholesterol Test Date : 16 bytes.
cholesterol Test result : 4 bytes.

AND SO ON.

b.) The result are stored in separate block: →

record length: 4 bytes.
pointer to address: 4 bytes.
pointer to patient history: 4 bytes.
birth Date: 10 bytes
ssn: 10 bytes.
patient ID: 10 bytes.
name: variable length.
~~patient history: variable length.~~
~~patient fees~~
address: variable length.
patient history: variable length.
pointer to cholestrol Test block: 4 bytes.

→ cholestrol Test Date: 16 bytes.
cholestrol Test Result: 4 bytes.
cholestrol Test Date: 16 bytes.
cholestrol Test Result: 4 bytes.
⋮ ⋮

AND SO ON.

PART 2 :→

2.1 a) For x:→

x: [300 - 320] Y: [500 - 550]
[320 - 340] [550 - 600]
[340 - 360] [600 - 650]
[360 - 380] [650 - 700]
[380 - 400] [700 - 750]

∴ 5 x 5
= 25

b) To find the nearest records we see in the range

For lower value ie 110
we look ~~from~~ in the range of:

$\boxed{80 - 140}$

For upper value ie 205
the nearest closest value would be :-
$\boxed{150 - 250}$

(i) [80 - 100] [150 - 200]
[100 - 120] [150 - 200]
[120 - 140] [150 - 200]

[80 - 100] [200 - 250]
[120 - 140] [200 - 250]

5 buckets are the nearest neighbours.

[100 – 120] [200 – 250] is already used in the question, so not counting it.

PART – 3 :→

3.1 a) $\pi_{AD} [R \bowtie \pi_{BD} (\sigma_{C=5} S)]$

There can be a variation as it was not specified whether the selection can be pushed down or not.

So, variation would be :→

$\pi_{AD} [\sigma_{C=5} (\pi_{ACD} (R \bowtie S))]$

b) $\sigma_{(C=2 \vee A=1)} [R \bowtie S]$

c) If we select all rows with C = 2 from S ~~from~~ & all rows with C = 1 from T

Then running a join (natural), there would be no output.

∴ Avoiding the query is best

∴ Not feasible.

PART - 4 :→

We are assuming :-

(i) Containment of value sets ie :→

The table with fewer distinct values for the join column, would join with some rows in the other table.

A attribute 'Y' in a relation R(..., Y) takes on a prefix of a fixed lists.

This would help use ~~the~~ estimate ~~of~~ the size ~~of~~ R(X, Y) ⋈ S(Y, Z).

- An attribute Y in a relation R(..., Y) always takes on a prefix of a fixed list of values : Y_1, Y_2, Y_3, Y_4.

4.1 →

~~T(W) = 100~~

~~S(W) = 30~~

~~V(W, A) = 1~~

~~V(W, B) = 1~~

~~V(W, C) = 100~~

PART 4:→

4.1 $$T(W) = \frac{T(R)}{V(R,A) \cdot V(R,B)}$$

$$= \frac{100000\cancel{0}}{2\cancel{0} \times 5\cancel{0}}$$

$$= 100$$

$$S(W) = 10 + 10 + 10$$
$$= 30 \text{ bytes}$$

$$V(W,A) = 1$$

$$V(W,B) = 1$$

$$V(W,C) = 100$$

4.2 $$T(Y) = \frac{T(W) \cdot T(S)}{\max(V(W,B), V(S,B)) \cdot \max(V(W,B), V(S,C))}$$

$$= \frac{100 \cdot 5000}{\max(1,100) \cdot \max(100, 200)}$$

$$= \frac{\cancel{100} \times \overset{25}{\cancel{5000}}}{\cancel{100} \times \cancel{200}}$$

$= \underline{\underline{25}}$

$S(Y) = 10 + 10 + 10 + 10$
$= \underline{\underline{40 \text{ bytes}}}$

$V(Y, A) = \underline{\underline{1}}$

$V(Y, B) = \underline{\underline{1}}$

$V(Y, C) = \underline{\underline{25}}$ $V(Y, D) = \underline{\underline{25}}$

4.3 :→ Finally taking the whole expression:→

$T(V) = \underline{\underline{25}}$

$S(V) = 10 + 10 + 10$
$= \underline{\underline{30 \text{ bytes}}}$.

$V(V, A) \to \underline{\underline{1}}$

$V(V, C) \to \underline{\underline{25}}$ $V(V, D) = \underline{\underline{25}}$

(c) R → A, B, C : each of 10 bytes long

S → A, D, E : A → 10 Bytes
D, E → 15 bytes.

R & S have → 64000 tuples each

R1 → 10 + 10 + 10 = 30 Bytes.
R2 → 10 + 15 + 15 = 40 Bytes

$$B(R) \rightarrow \frac{30 \times 64000}{6400}$$
$$= 300 \text{ Blocks}$$

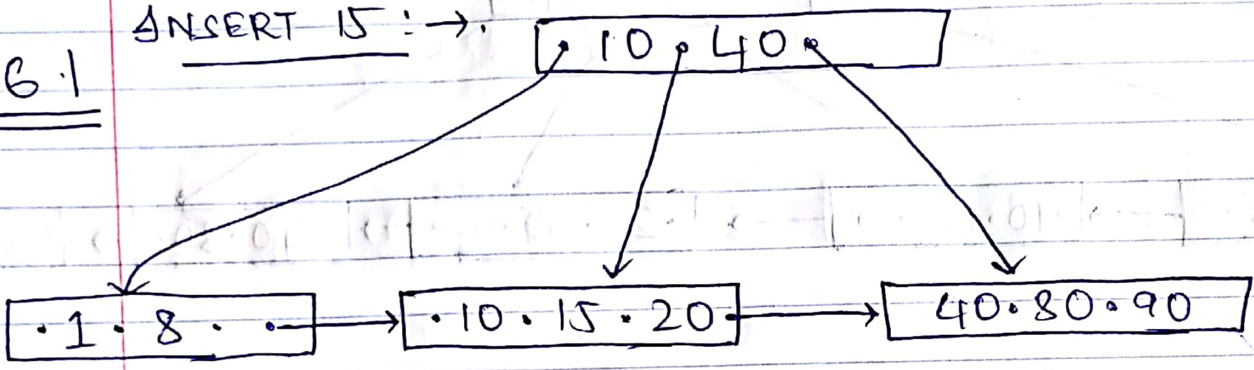
$$B(S) \rightarrow \frac{40 \times 64000}{6400}$$
$$= 400 \text{ blocks.}$$

∴ Total cost = 3 (B(R) + B(S))
= 3 (300 + 400)
= 2100 blocks

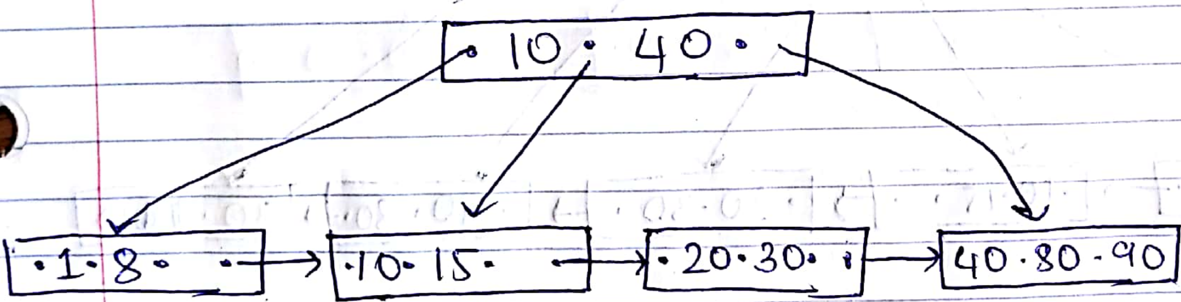
PART-6 :→

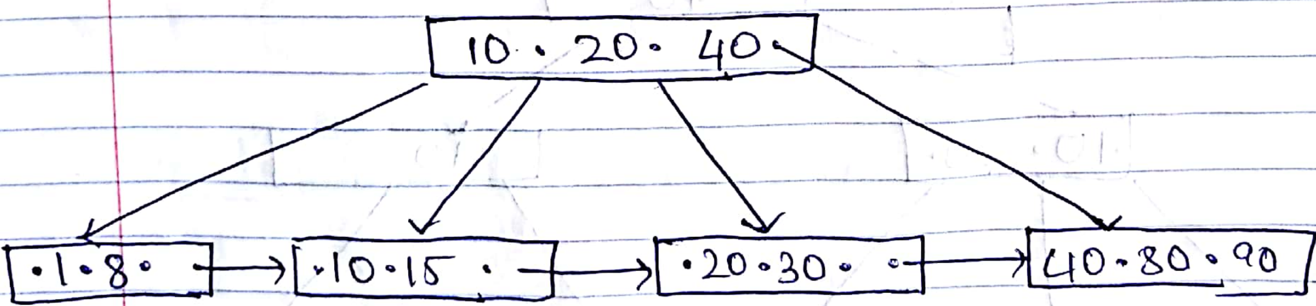
6.1

INSERT 15 :→

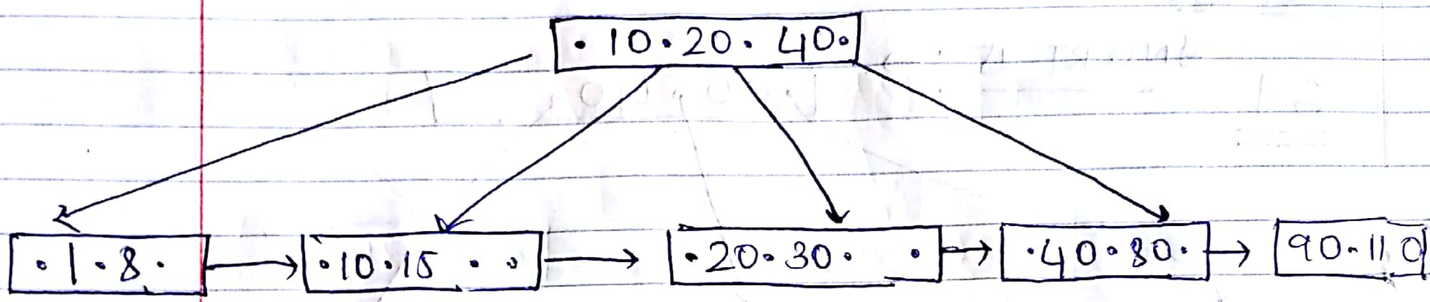


INSERT 30 :→

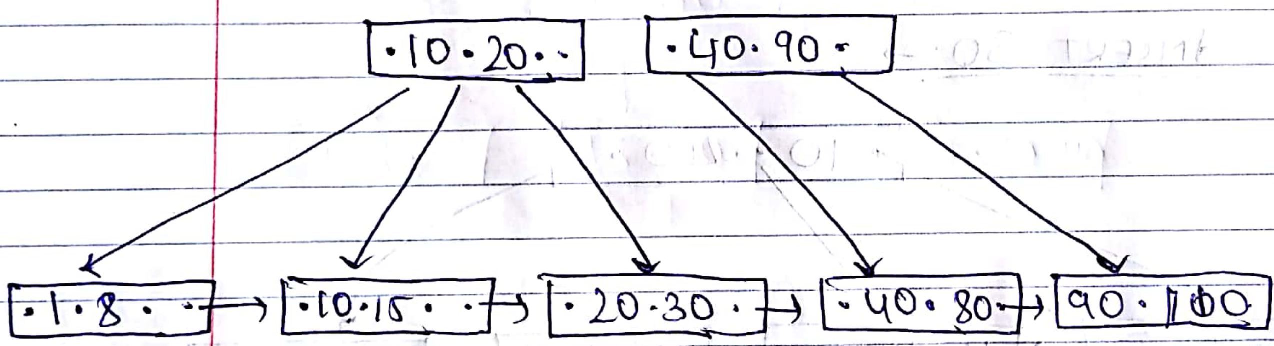




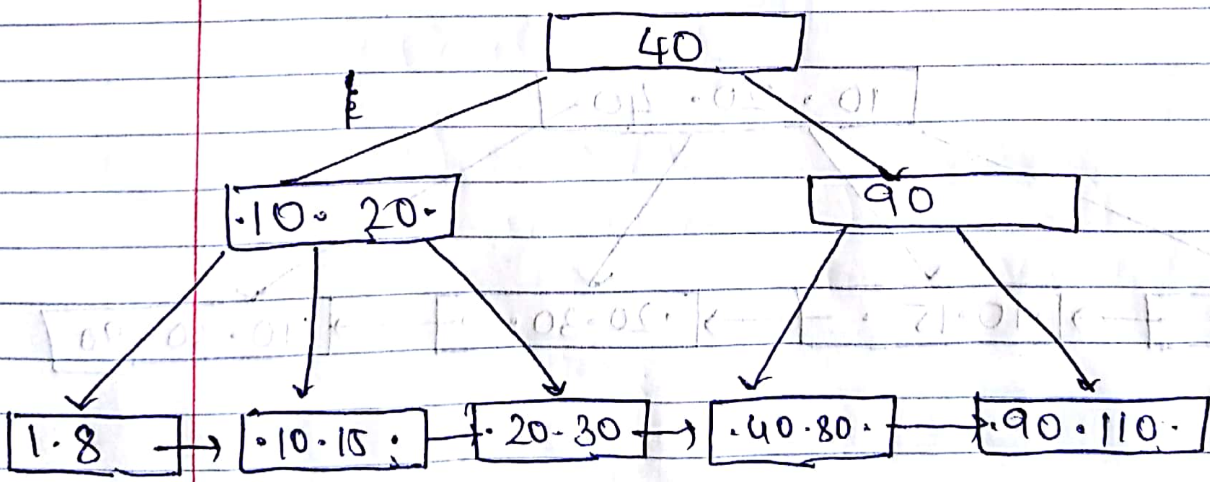
Insert 110:→



• 10 • 20 • 40 •
• 1 • 8 •
• 10 • 15 • •
• 20 • 30 • •
• 40 • 80 •
90 • 110

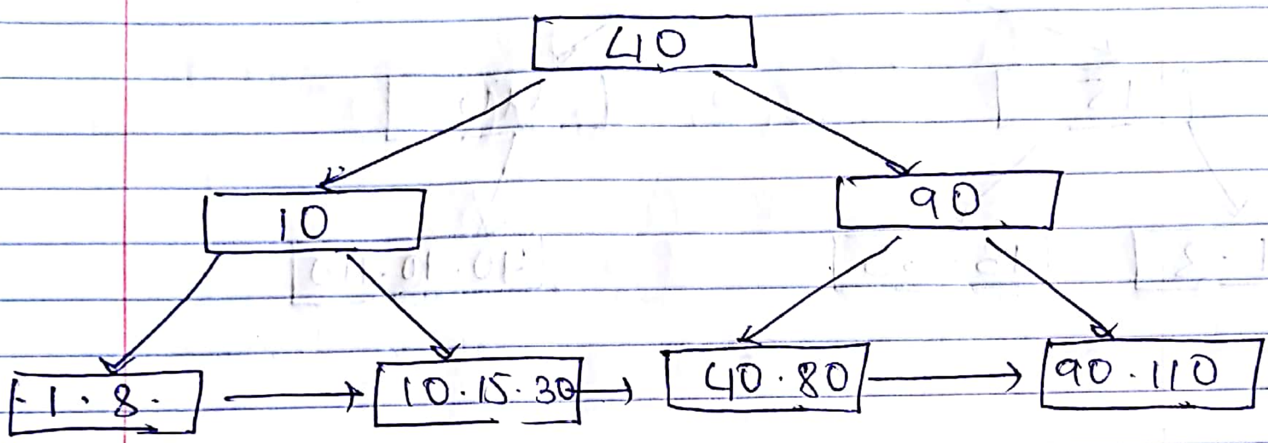


• 10 • 20 • •
• 40 • 90 •
• 1 • 8 • •
• 10 • 15 • •
• 20 • 30 •
• 40 • 80 •
90 • 100

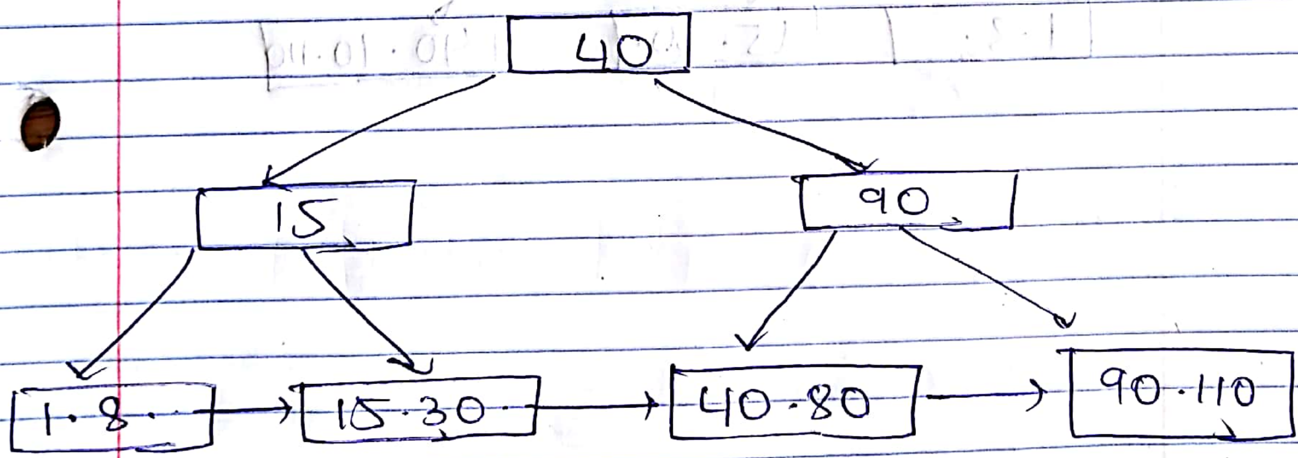


40
• 10 • 20 •
90
1 • 8
• 10 • 15 •
• 20 • 30
• 40 • 80 •
• 90 • 110 •

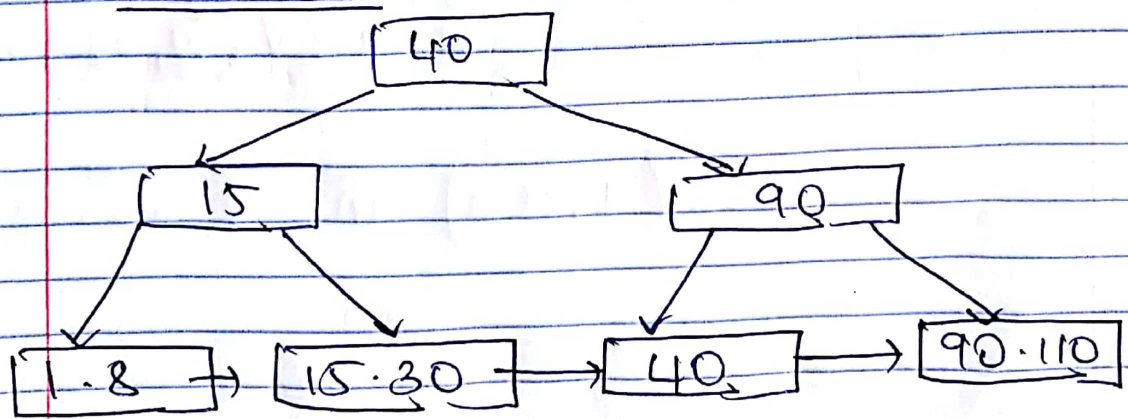
DELETE 20 :→



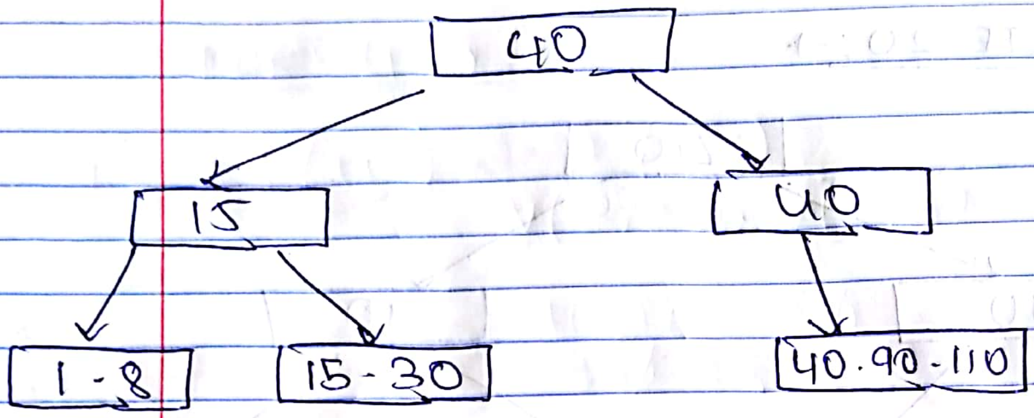
Delete 10 :→

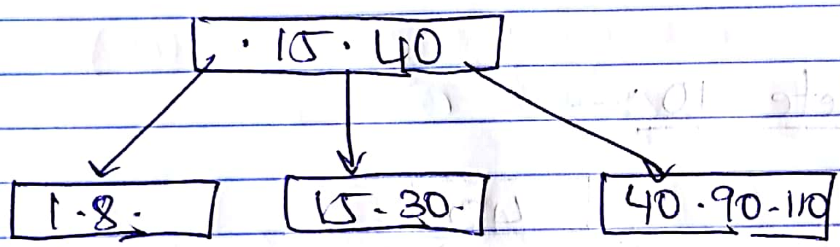


Delete 80 :→



Scanned by CamScanner


40
15
40
1 - 8
15 - 30
40 . 90 - 110


. 15 . 40
1 . 8 .
15 - 30 .
40 . 90 - 110