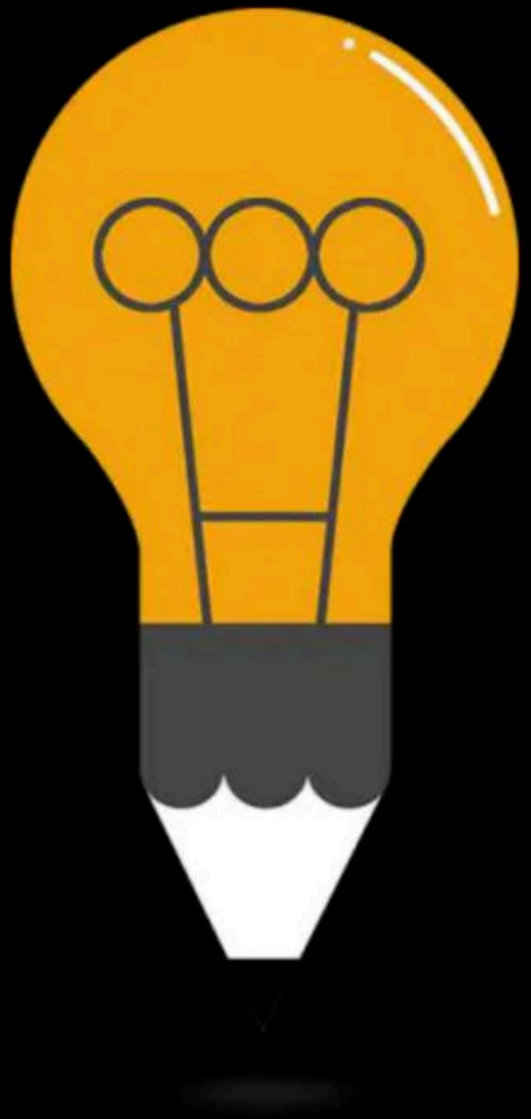
unacademy

Doubt Clearing Session

Complete Course on Database Management System

Vishvadeep Gothi • Lesson 32 • July 28, 2023

unacademy



DBMS

Indexing 4

By: Vishvadeep Gothi

unacademy

Indexing Techniques

- Primary Indexing
- Clustering Indexing
- Secondary key Indexing
- Secondary non-key Indexing

unacademy

Indexing Techniques

- Primary Indexing
- Clustering Indexing
- Secondary key Indexing
- Secondary non-key Indexing

unacademy

Primary Indexing

- Indexing done on primary key or any super key
- Data must be ordered on index
- Its always sparse index

unacademy

Clustering Indexing

- Indexing done on non-key field
- Data must be ordered on index field
- Sparse or dense

⇒ Indexing done on each unique value.

Sparse → repeated

dense → all unique values

unacademy

Secondary key Indexing

- Indexing done on primary key or any super key
- Data must not be ordered on index field
- It is dense index

unacademy

Secondary Non-key Indexing

- Indexing done on non-key field
- Data must not be ordered on index field
- It is sparse index

↳ indexing is done on each unique value

Primary Index

Rno.	Block Pointer
1	B1
5	B2

Rno.	Block Pointer
9	B3
13	B4

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

13	C
14	S
15	H
16	A

unacademy

Primary Index

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

Select * from Students

where rno=1

unacademy

Primary Index

1	A
2	B
3	C
4	D

5	A
6	B
7	E
8	W

9	V
10	D
11	I
12	A

Select * from Students

where rno=12

unacademy

Question

Consider a database file of 65536 records, each record of size 64 bytes. Key field is 10 bytes and block pointer size is 22 bytes. Assume that block size is 256 bytes.

1. The number of blocks required to store file?
2. The number of blocks required to store the index file for primary indexing?

Clustering Index

S_Name	Block Pointer
A	B1
B	B2
B	B3
E	B4

1	A	B1
2	A	
3	A	
4	B	
5	B	B2
6	B	
7	B	
8	B	
9	B	B3
10	B	
11	C	
12	D	
13	E	B4
14	F	
15	F	
16	G	

Clustering Index

S_Name	Block Pointer
A	B1
B	B1
C	B3
D	B3
E	B4
F	B4
G	B4

1	A	B1
2	A	
3	A	
4	B	
5	B	B2
6	B	
7	B	
8	B	
9	B	B3
10	B	
11	C	
12	D	
13	E	B4
14	F	
15	F	
16	G	

unacademy

Question

DB File size 1G records

Record size = 64bytes

Block size = 4096bytes

Index field = 10 bytes

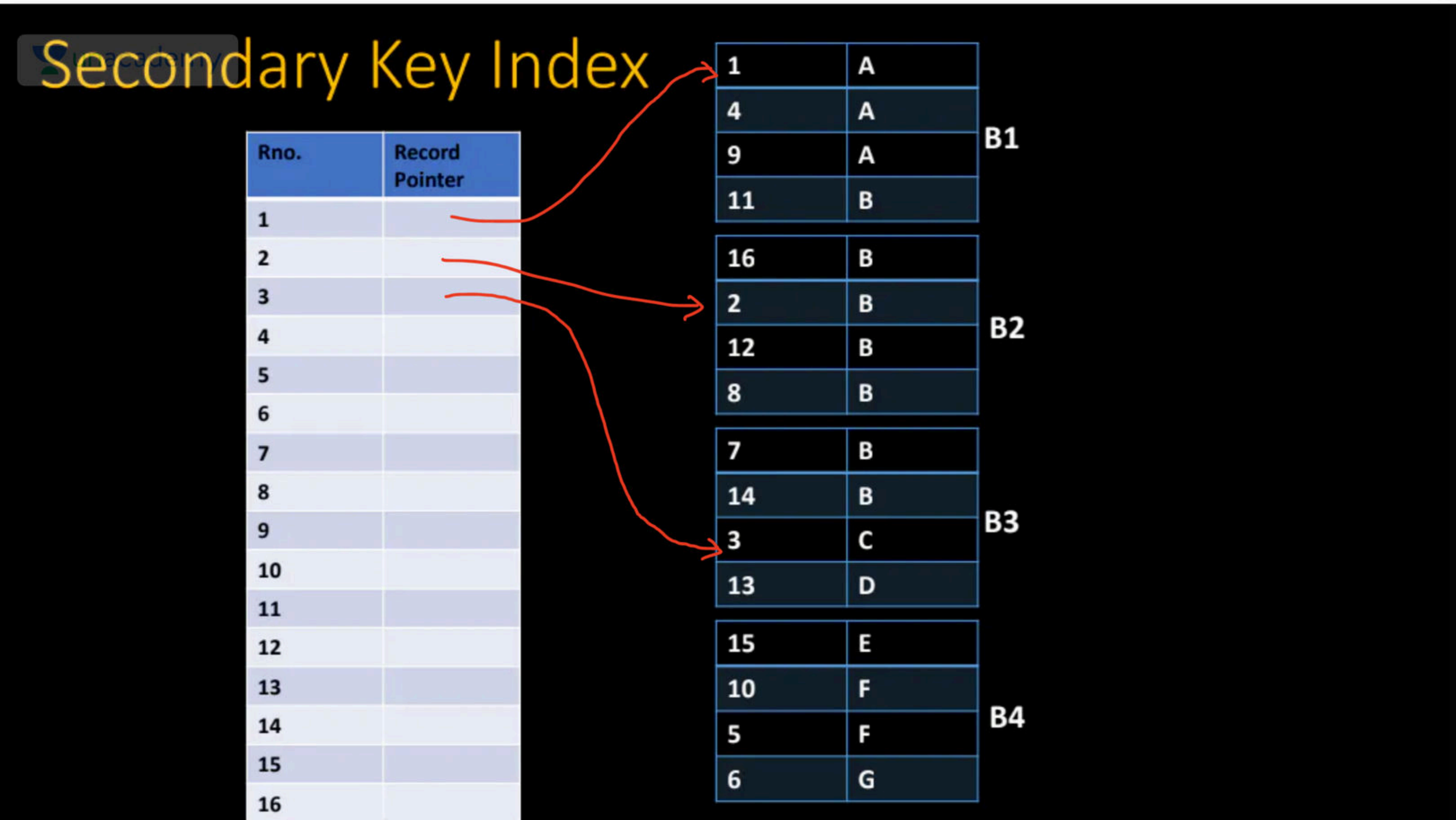
Block pointer size = 22 bytes

Number of distinct values in index file = 16384

Indexing is done on non-key, data is ordered on non-key and indexing is done for each distinct non-key value

Secondary Key Index

Rno.	Record Pointer
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	



1	A
4	A
9	A
11	B

B1

16	B
2	B
12	B
8	B

B2

7	B
14	B
3	C
13	D

B3

15	E
10	F
5	F
6	G

B4

unacademy

Question

DB File size 2^{40} bytes

Record size = 128bytes

Block size = 4096bytes

Index key field = 10 bytes

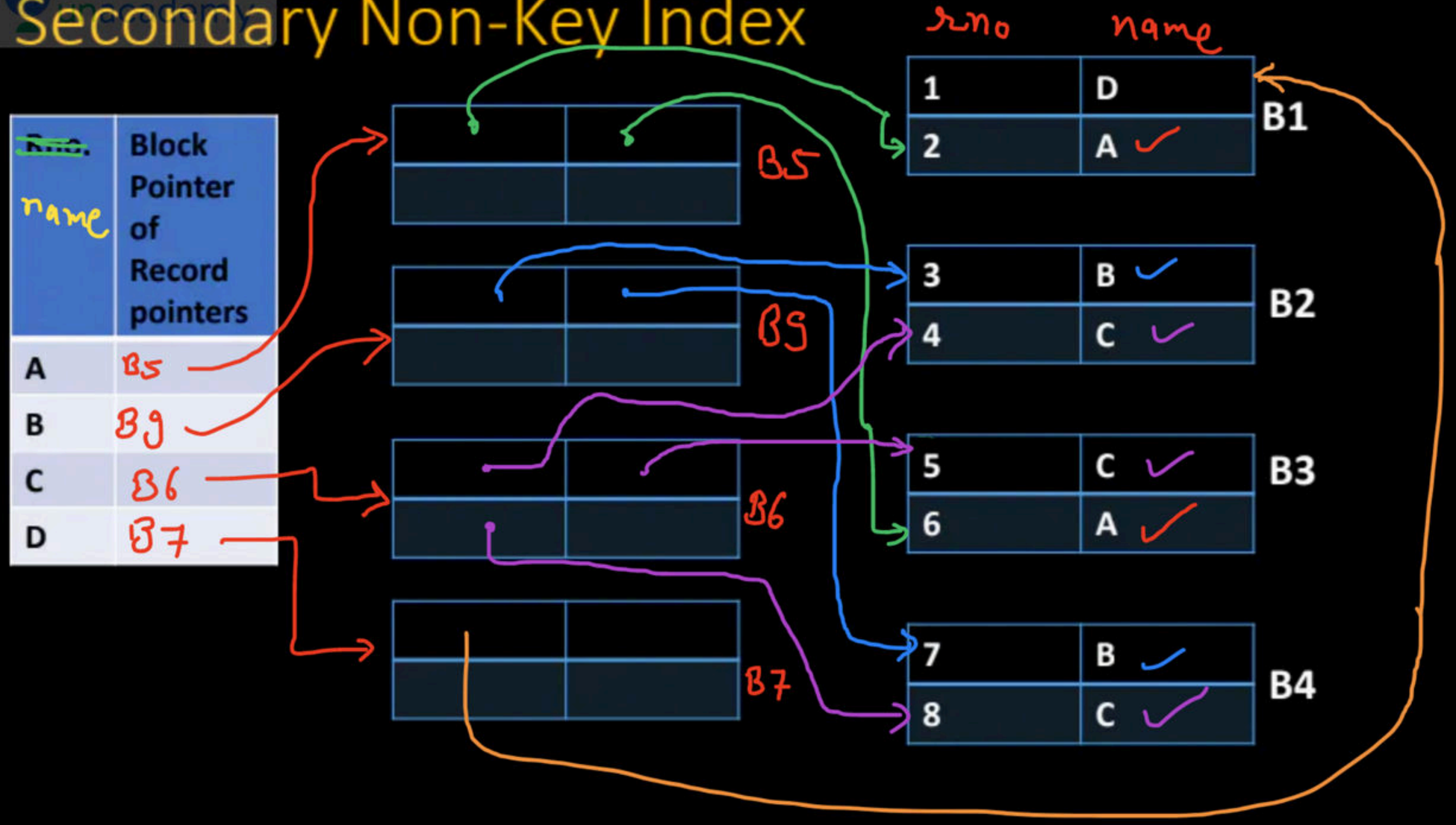
Record pointer size = 22 bytes

unspanned file organization

Indexing is done on key; data is unordered on key and indexing is done for each key value

1. Number of blocks required to store database file
2. Number of blocks required to store index file

Secondary Non-Key Index



B5 is index block for value A .

unacademy

Recap

Ordering	Key or Non Key	Type
Ordered	Key	Primary
Ordered	Non-Key	clustering
Unordered	Key	Sec. key
Unordered	Non-Key	sec non-key

unacademy

Question

Homework Question 1

DB File size 10^8 records

Record size = 400bytes

Block size = 4096bytes

Index key field = 16 bytes

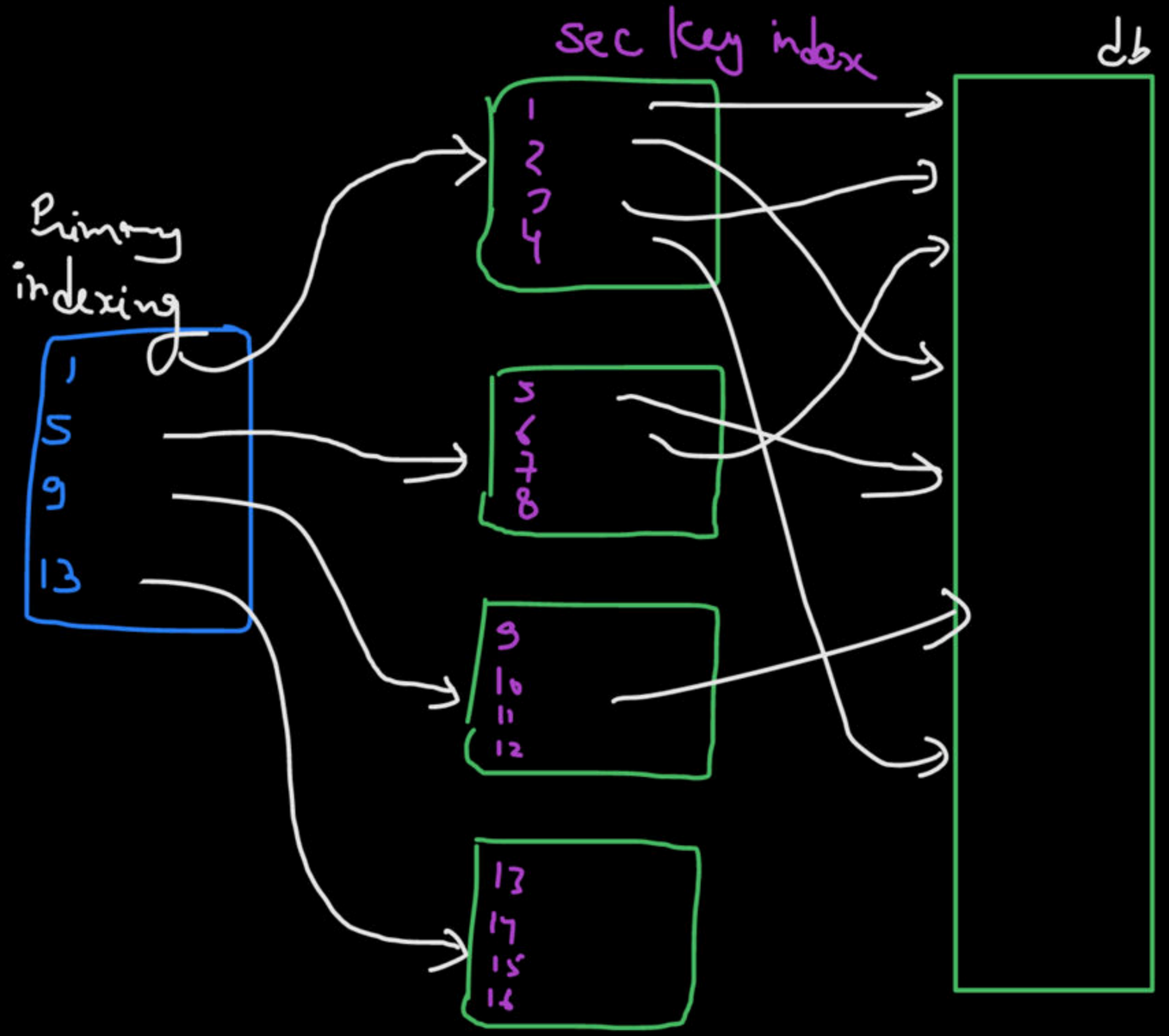
index pointer size = 4 bytes

unspanned file organization

Indexing is done on key; data is unordered on key and indexing is done for each key value

1. Number of blocks required to store database file
2. Number of blocks required to store index file

Multilevel Indexing



w/o multilevel index

⇓

To search indexes no. of blocks to be accessed = $\log_2 b$

b = no. of blocks required to store indexes

[one access extra to access db record]

with n-level multilevel indexing :-

To access index (rec. pointer), no. of blocks accessed = n

[one block access extra to access db record]

unacademy

Question GATE-2008

Consider a file of 16384 records. Each record is 32 bytes long and its key field is of size 6 bytes. The file is ordered on a non-key field, and the file organization is unspanned. The file is stored in a file system with block size 1024 bytes, and the size of a block pointer is 10 bytes. If the secondary index is built on the key field of the file, and a multi-level index scheme is used to store the secondary index, the number of first-level and second-level blocks in the multi-level index are respectively.

(A) 8 and 0

(B) 128 and 6

(C) 256 and 4

(D) 512 and 5

no. of index rec. needed = 16384 = 2^{14}

index rec. size = 6 + 10 = 16 bytes

no. of index rec. per block = $\frac{1024\ B}{16\ B} = 2^6$

no. of blocks, for 2^{14} indexes = $\frac{2^{14}}{2^6} = 2^8$

for second level ⇒ Primary indexing

no. of index records = no. of blocks = 2^8

no. of blocks, to store 2^8 indexes on second level = $\frac{2^8}{26} = 4$

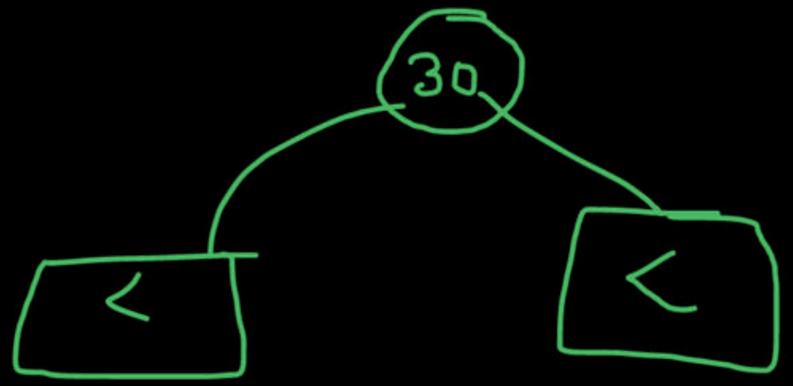
unacademy

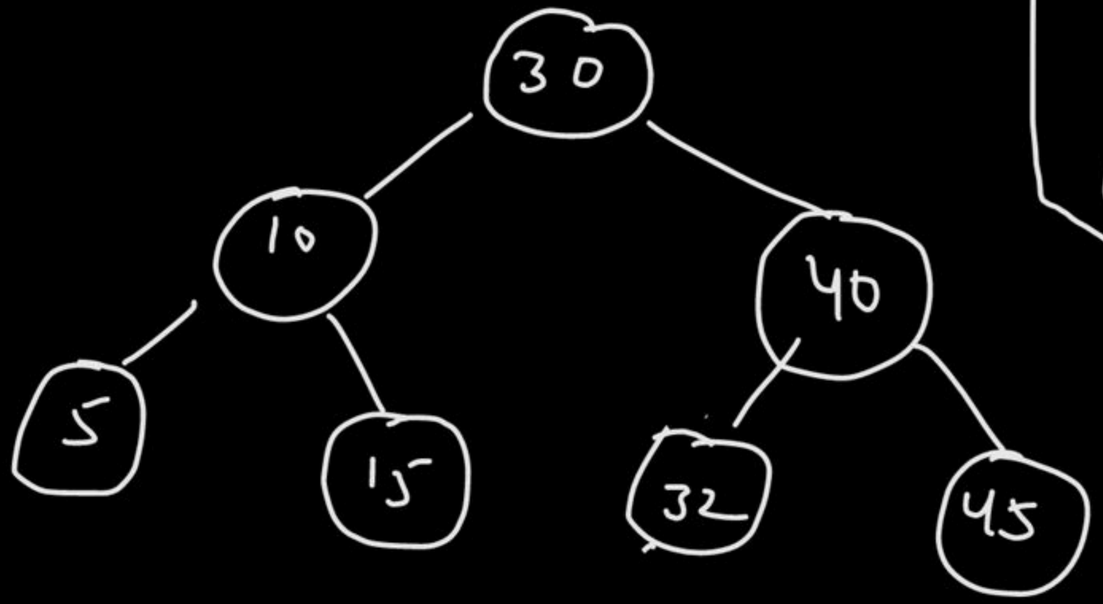
B-Tree

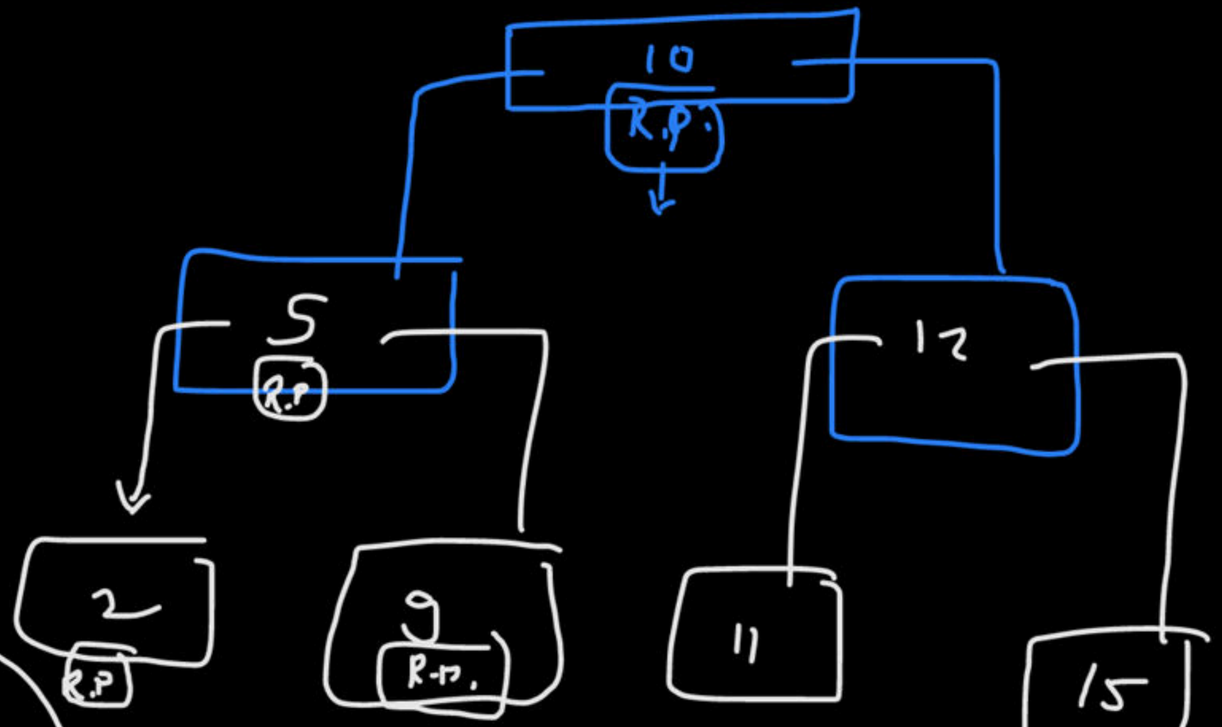
- Tree based indexing
- Dynamic Indexing technique
- Based on insertion and deletion, the tree automatically adjusted
- Self balancing search tree

unacademy

Binary Search Tree







as no. of elements in BST increases ⇒ Tree grows vertically and it's height increases.

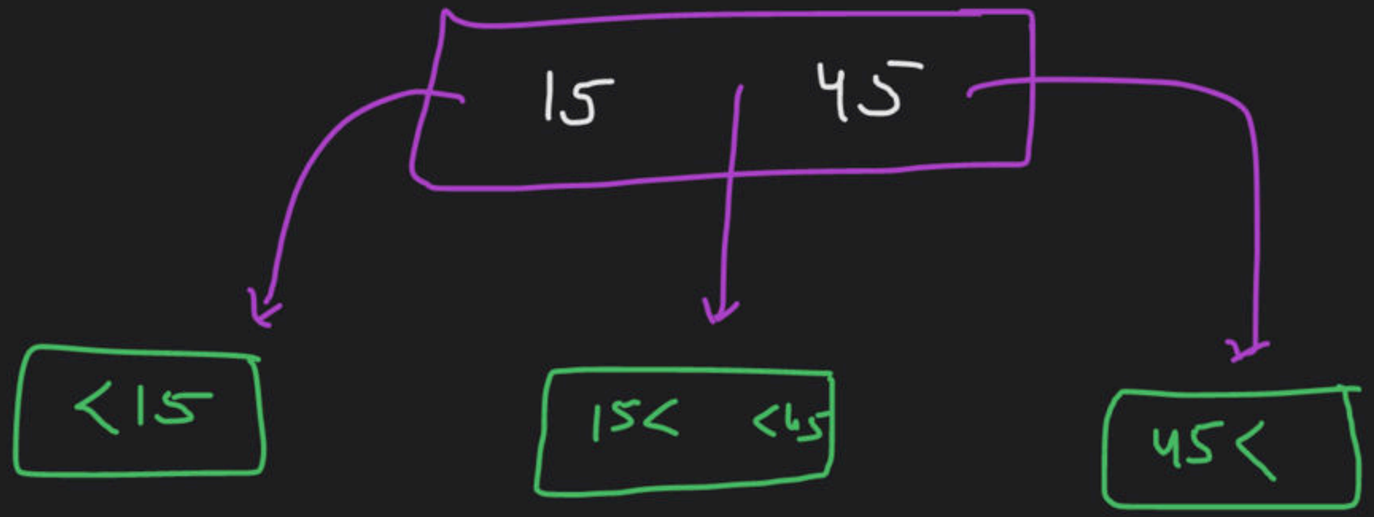
hence even for searching when $\log_2 n$ time required, that too becomes large.

Soln ⇒ make a tree which can grow horizontally and vertically also.

unacademy

Ex:-

3-ary tree ⇒ each node can have 3 children ⇒ 2 keys



unacademy

B-Tree

An order-p B-tree:

1. Every node other than root should have atleast $\left\lceil \frac{p}{2} - 1 \right\rceil$ ~~nodes~~ keys or $\left\lceil \frac{p}{2} \right\rceil$ pointers
2. In every node there are atmost (p-1) keys and p tree pointers
3. Root can have minimum 1 keys
4. All leaves appear on the same level

Ex:- order-3 B-Tree

	max	min.
keys	2	1
pointers	3	2

unacademy

Ex: order-4 B-Tree

	max	min
keys	3	1
pointers	4	2

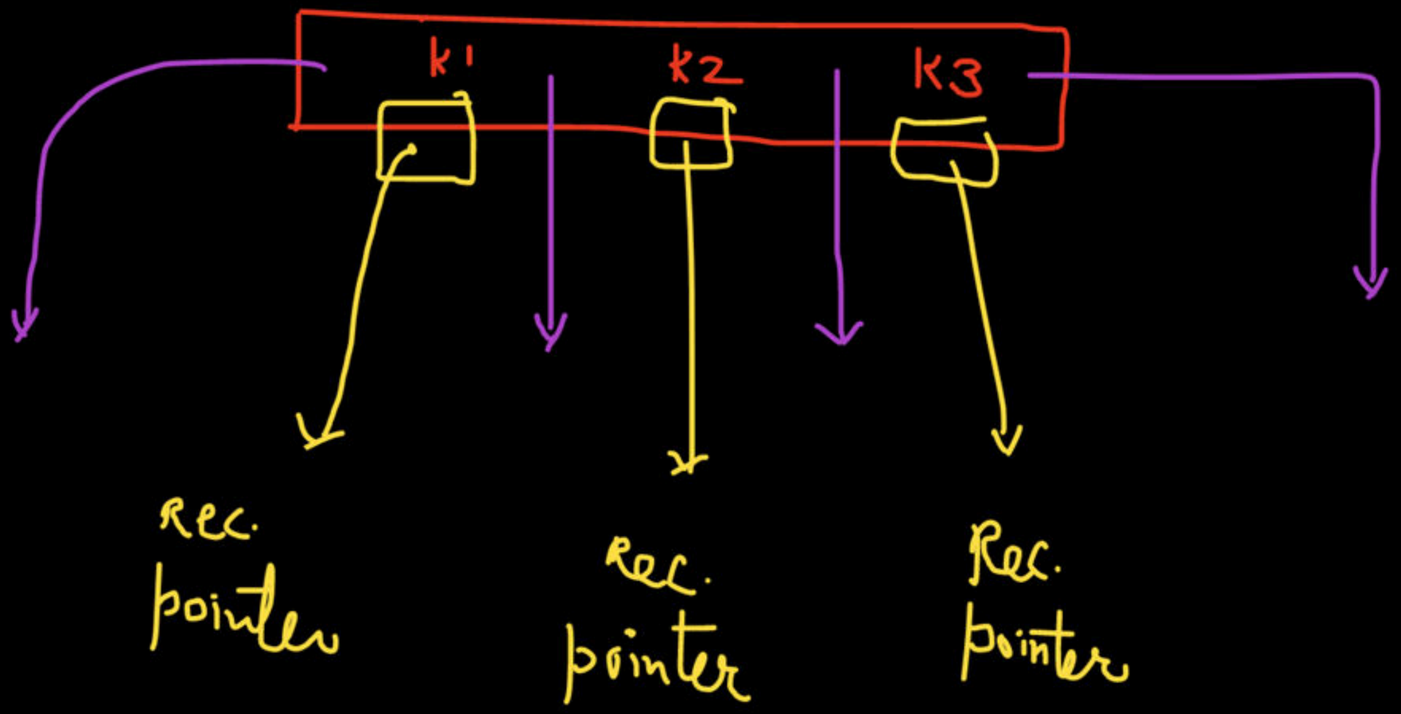
order-5 B-Tree

	max	min
keys	4	2
pointers	5	3

unacademy

B-Tree Node Structure

- Key
- Record Pointer
- Tree Pointer



unacademy

Insertion in B-Tree

- B-tree of order-3
- Insert keys 1, 2, 3, 4, 5, 6, 7

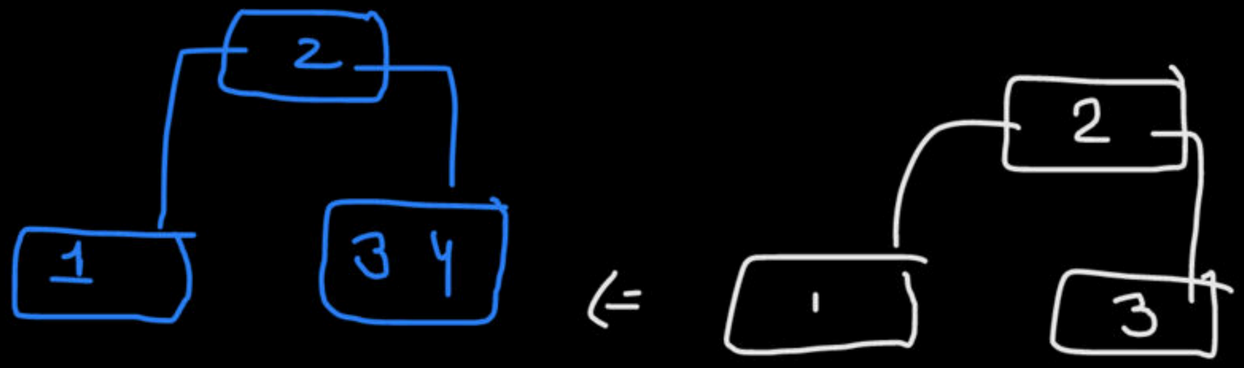
↳ insertion is always done on leaf node

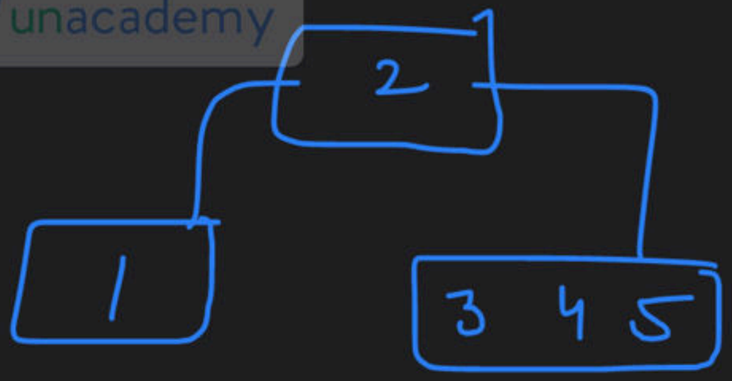
	min	max
keys	1	2

root

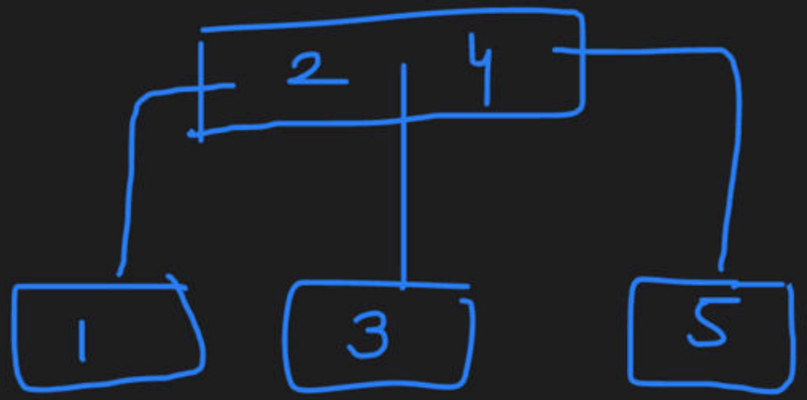
1 ⇒ 1 2 ⇒ 1 2 3

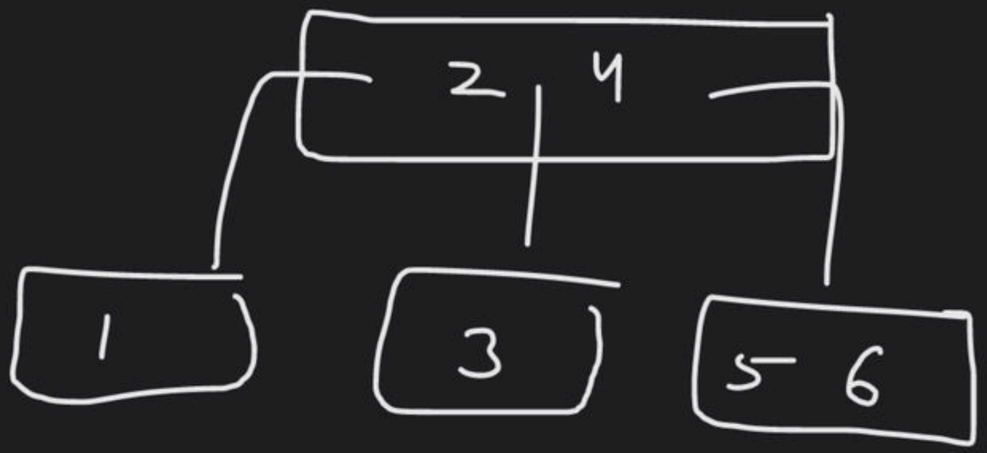
⇓ node split

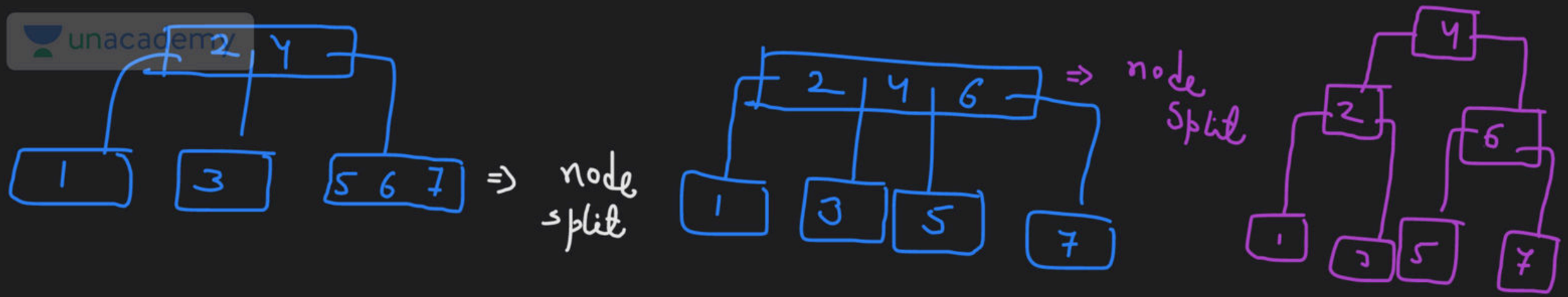




⇒ node split



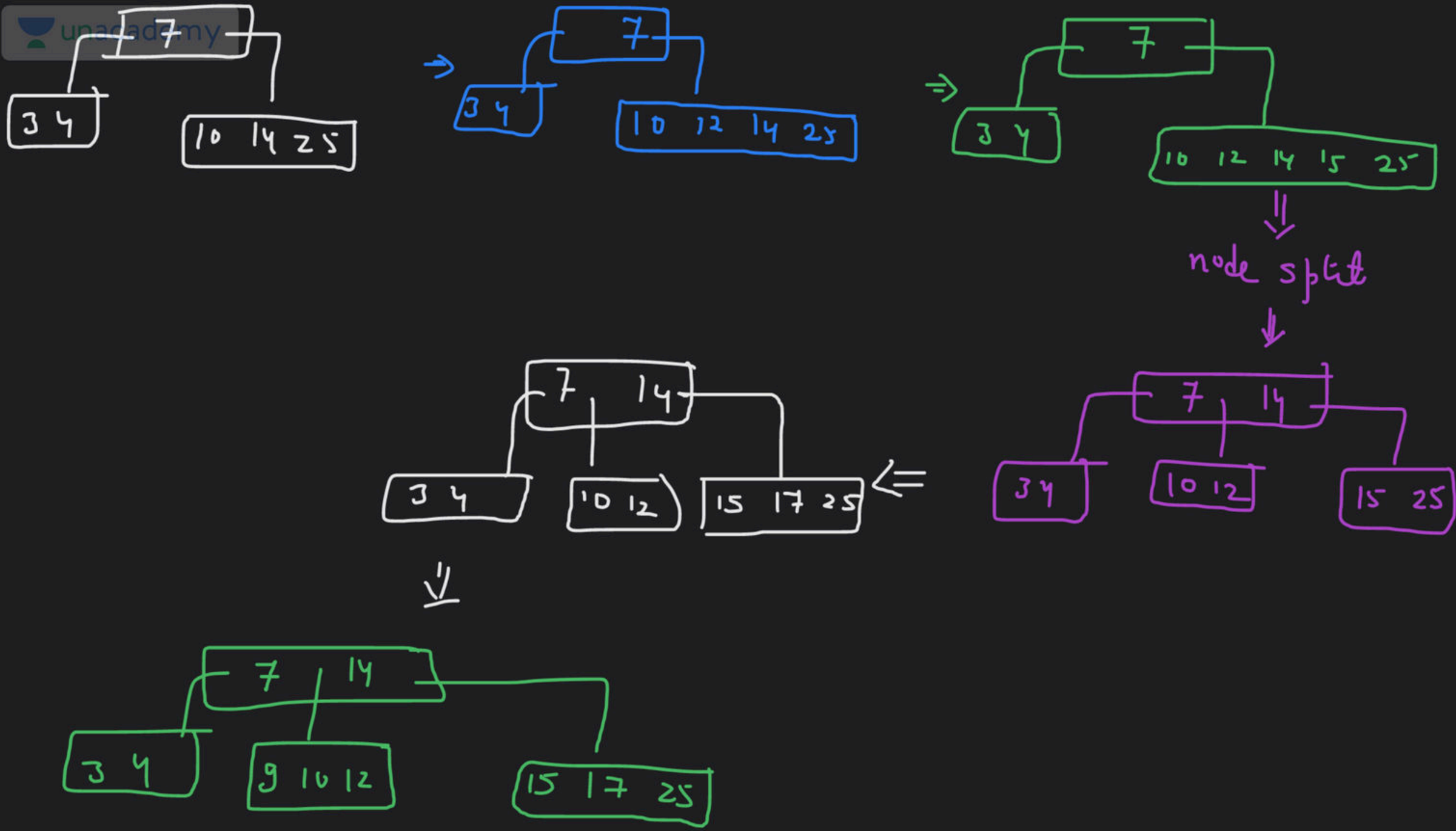


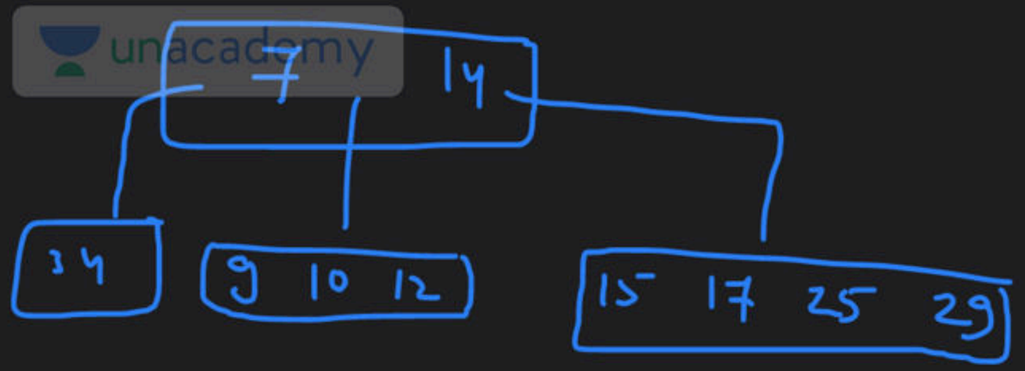

2
4
1
3
5
6
7
node
split
2
4
6
1
3
5
7
node
Split
4
2
6
1
3
5
7

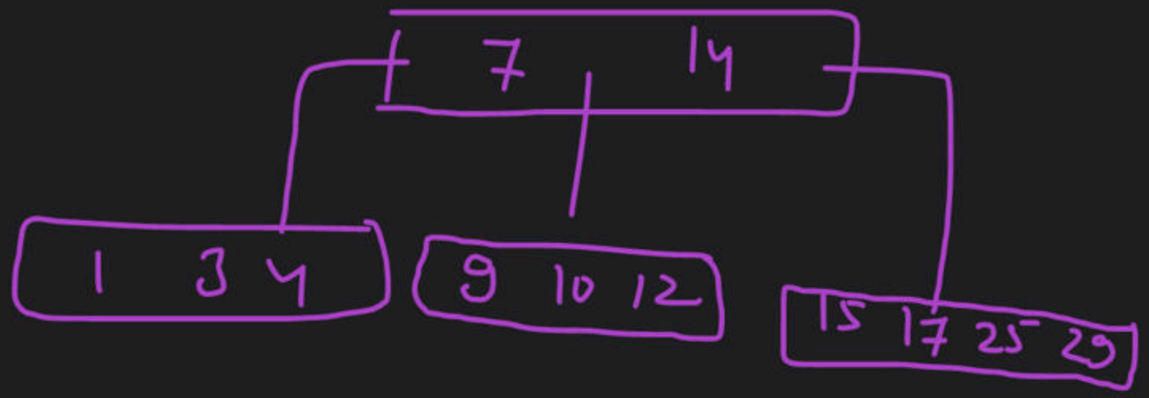
Insertion in B-Tree

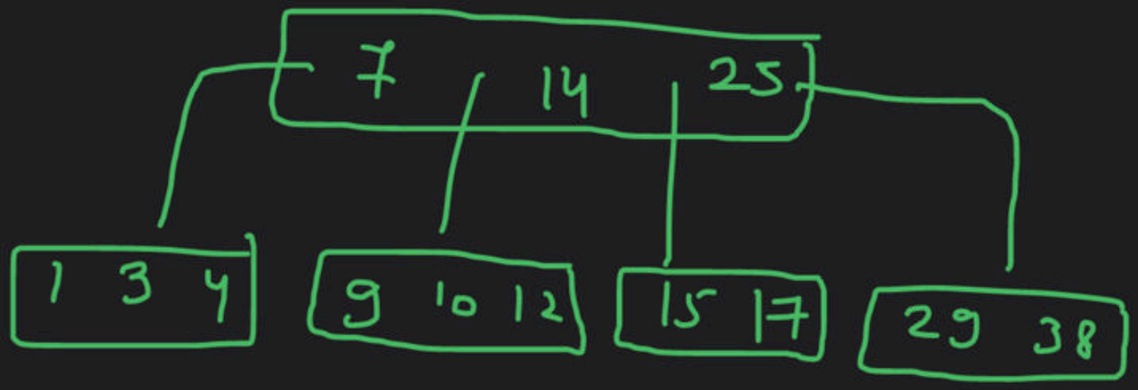
- B-tree of order-5
- Insert keys 7, 4, 14, 25, 3, 10, 12, 15, 17, 9, 29, 1, 38, 8, 11

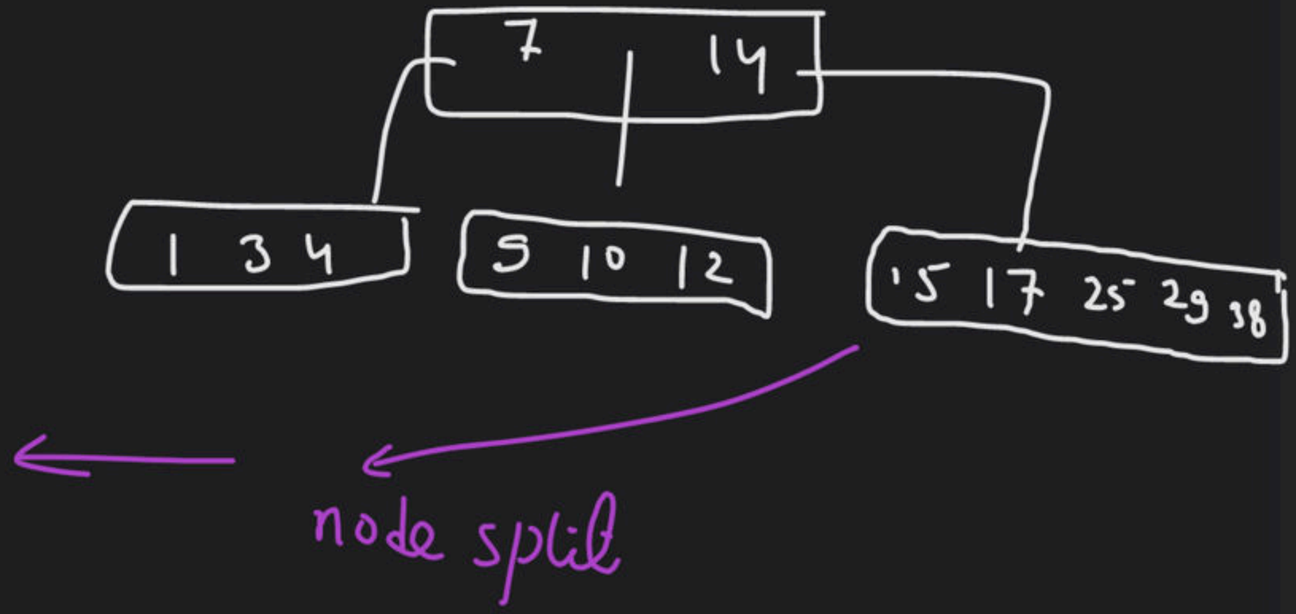
	max	min
keys	4	2

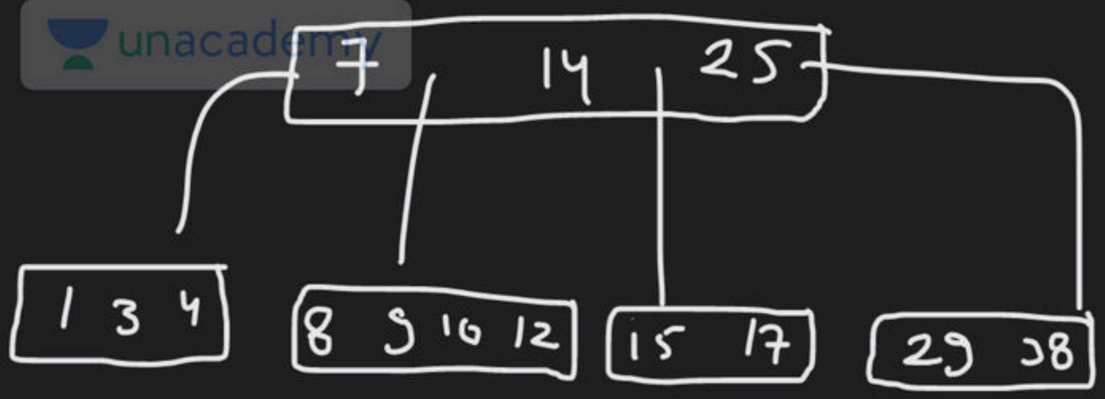

7
3 4
10 14 25
7
3 4
10 12 14 25
7
3 4
10 12 14 15 25
node split
7 14
3 4
10 12
15 25
7 14
3 4
10 12
15 17 25
7 14
3 4
9 10 12
15 17 25

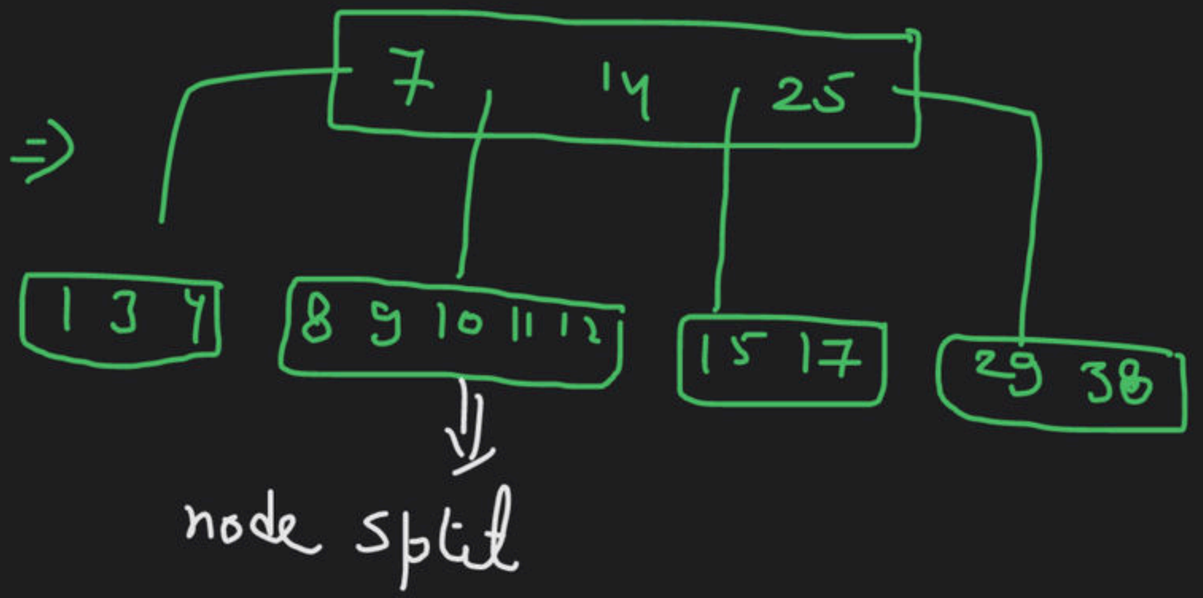

7
14
3 4
9 10 12
15 17 25 29

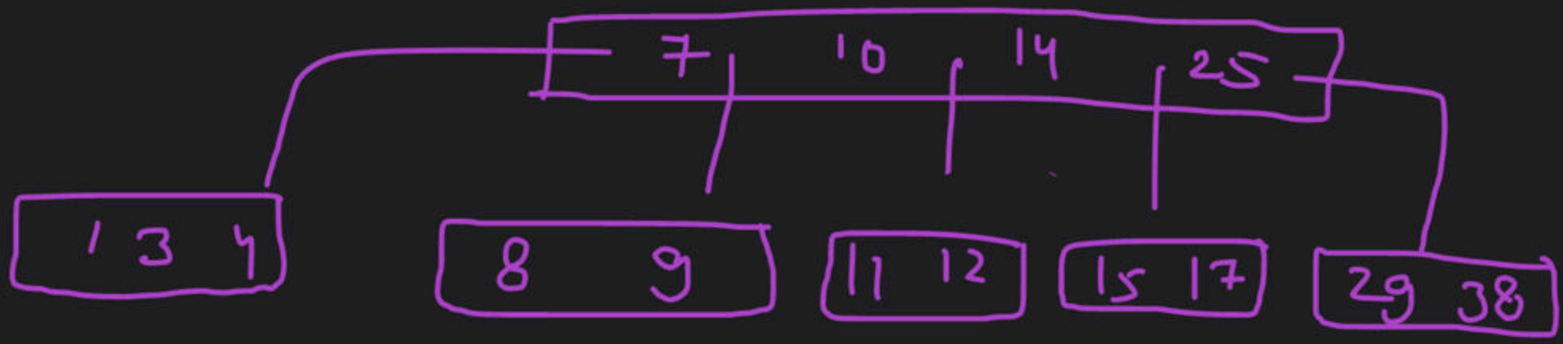

7
14
1 3 4
9 10 12
15 17 25 29


7
14
25
1 3 4
9 10 12
15 17
29 38


7
14
1 3 4
9 10 12
15 17 25 29 38
node split


7
14
25
1 3 4
8 9 10 12
15 17
29 38


7
14
25
1 3 4
8 9 10 11 12
15 17
29 38
node split


7
10
14
25
1 3 4
8 9
11 12
15 17
29 38

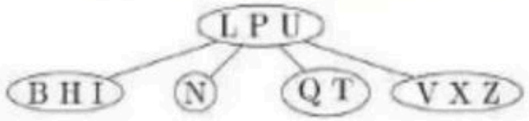
unacademy

Insertion in B-Tree

- B-tree of order-3
- Insert keys 14, 3, 5, 10, 35, 40, 1, 37

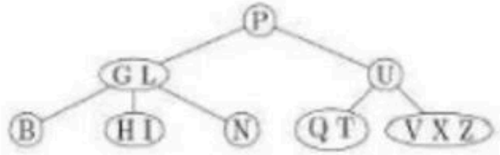
Question GATE-2003

Consider the following $2-3-4$ tree (i.e., B-tree with a minimum degree of two) in which each data item is a letter. The usual alphabetical ordering of letters is used in constructing the tree.

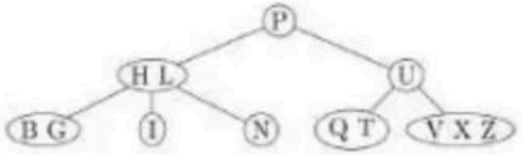


What is the result of inserting G in the above tree?

A.



B.



C.

D. None of the above

unacademy

Insertion in B-Tree

- B-tree of order-4
- Insert keys 15, 5, 8, 22, 10, 1

unacademy

Insertion in B-Tree

More split in left-biasing or right-biasing?

unacademy

Question GATE-2008

A B-tree of order 4 is built from scratch by 10 successive insertions. What is the maximum number of node splitting operations that may take place?

(A) 3

(B) 4

(C) 5

(D) 6

unacademy

Question

A B-tree of order 4 is built from scratch by successive insertions of following keys in the given order.

10, 5, 14, 10, 3, 6, 30, 27, 9

What is the number of root node splitting operations that may take place with right biasing?

unacademy

Question GATE-2005

A B-Tree used as an index for a large database table has four levels including the root node. If a new key is inserted in this index, then the maximum number of nodes that could be newly created in the process are?

unacademy

Question GATE-2004

Consider a table T in a relational database with a key field K. A B-tree of order p is used as an access structure on K, where p denotes the maximum number of tree pointers in a B-tree index node. Assume that K is 10 bytes long; disk block size is 512 bytes; each data pointer PD is 8 bytes long and each block pointer PB is 5 bytes long. In order for each B-tree node to fit in a single disk block, the maximum value of p is?

Practical Implementation of Node on Blocks

What is maximum order in B-tree?

unacademy

Question

Key size = 16 bytes

Block pointer size = 32 bytes

Record pointer size = 48 bytes

Block size = 8192 bytes

If a B-tree of order-p is implemented, then what is the maximum value of p?

unacademy

Height of the B-tree

P-order B-tree

Total nodes = n

$$H_{min} = \lceil \log_p(n+1) - 1 \rceil$$

$$H_{max} = \left\lfloor \log_{\lceil \frac{p}{2} \rceil} \frac{n+1}{2} \right\rfloor$$

unacademy

Deletion in B-Tree

2 Cases:

1. Deletion in leaf
2. Deletion in internal node

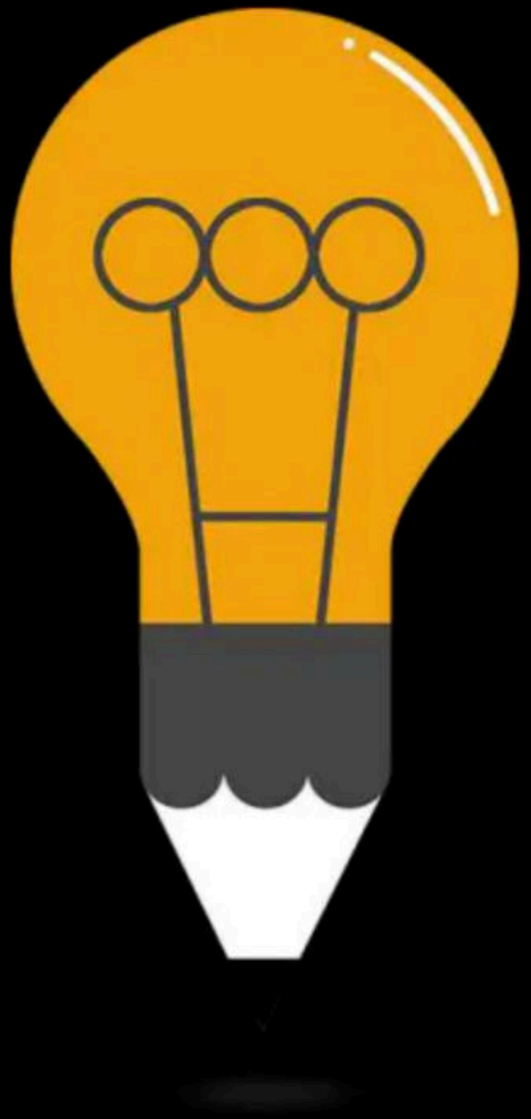
Deletion in B-Tree: Deletion in Leaf

1. After deletion if no violation of min keys, then no changes in tree
2. If violation of min keys, then borrow key from sibling (rotation through parent).
3. If borrow from sibling can’t be possible then merge the node with sibling and pull down the anchor key from parent.

Deletion in B-Tree: Deletion in Internal Node

1. Replace the deleted value with inorder successor or inorder predecessor
2. Now follow the rule of deletion of key from leaf node

unacademy



DBMS

Indexing DPP

By: Vishvadeep Gothi

unacademy

Question 2

DB File size 56000 records

Record size = 146 bytes

Block size = 4096bytes

Index key field = 12 bytes

index pointer size = 14 bytes

unspanned file organization

Indexing is done on key; data is unordered on key and indexing is done for each key value

1. Number of blocks required to store database file
2. Number of blocks required to store index file

unacademy

Question 3

DB File size 68734 records

Record size = 211 bytes

Block size = 8192 bytes

Index key field = 20 bytes

index pointer size = 41 bytes

spanned file organization

Indexing is done on key; data is ordered on key and primary indexing is done

1. Number of blocks required to store database file
2. Number of blocks required to store index file

unacademy

Question 4

DB File size 25400 records

Record size = 72 bytes

Block size = 1024 bytes

Index non-key field = 23 bytes

Number of non-key unique values = 18763

index pointer size = 29 bytes

unspanned file organization

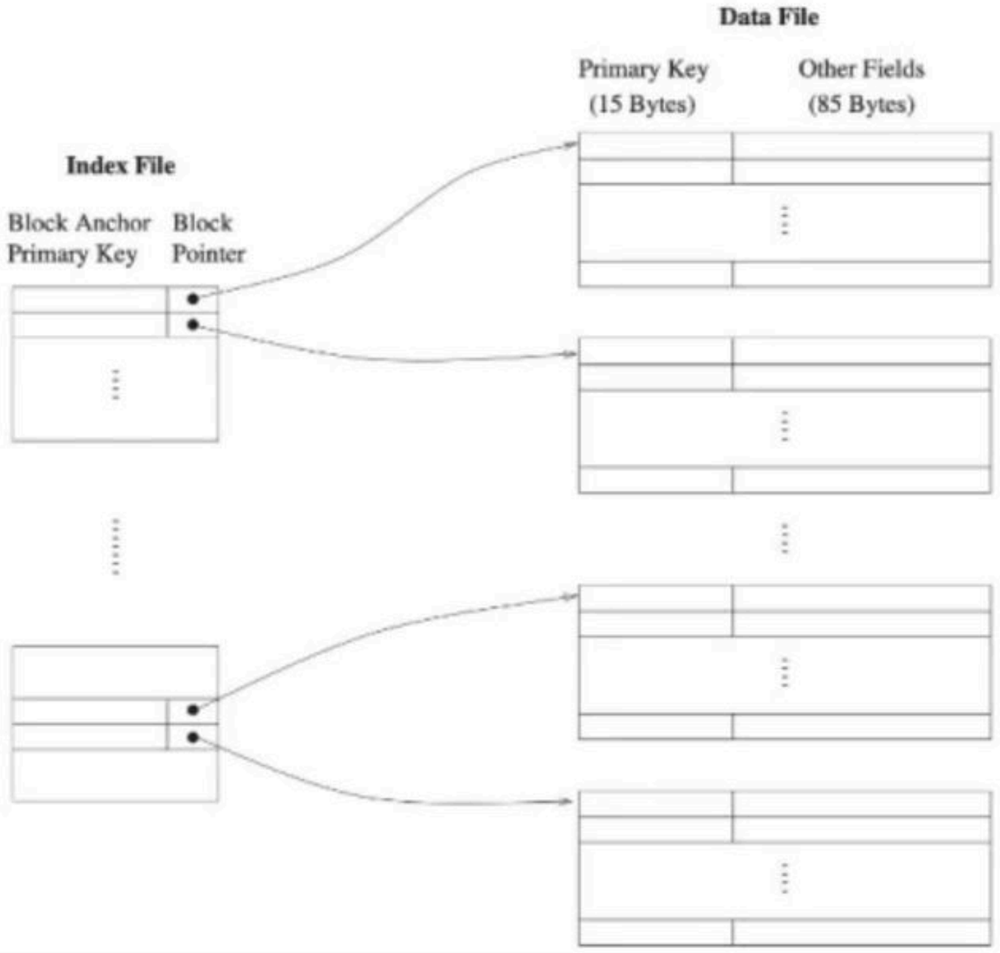
Indexing is done on non-key; data is ordered on non-key and indexing done on each unique value of non-key

1. Number of blocks required to store database file
2. Number of blocks required to store index file

unacademy

Question GATE-2023 5

Consider a database of fixed-length records, stored as an ordered file. The database has $25,000$ records, with each record being 100 bytes, of which the primary key occupies 15 bytes. The data file is block-aligned in that each data record is fully contained within a block. The database is indexed by a primary index file, which is also stored as a block-aligned ordered file. The figure below depicts this indexing scheme.



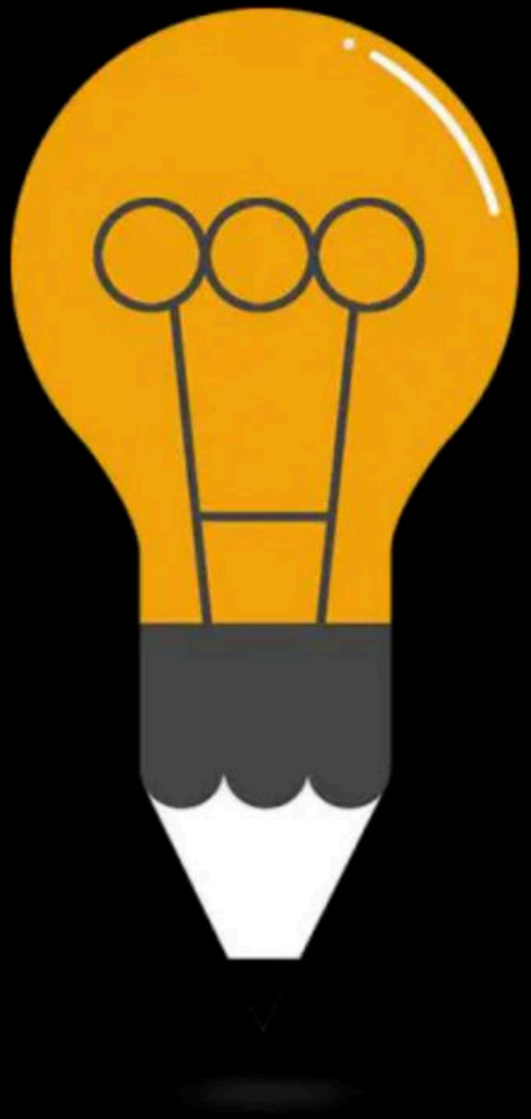
unacademy

Question GATE-2023 continue

Suppose the block size of the file system is 1024 bytes, and a pointer to a block occupies 5 bytes. The system uses binary search on the index file to search for a record with a given key. You may assume that a binary search on an index file of b blocks takes $\lceil \log_2 b \rceil$ block accesses in the worst case.

Given a key, the number of block accesses required to identify the block in the data file that may contain a record with the key, in the worst case, is _____.

unacademy



Transaction PYQs

By: Vishvadeep Gothi

Question GATE-2004

Which level of locking provides the highest degree of concurrency in a relational database ?

A. Page
B. Table
C. Row
D. Page, table and row level locking allow the same degree of concurrency

unacademy

Question GATE-1999

For the schedule given below, which of the following is correct:

1	Read A	
2		Read B
3	Write A	
4		Read A
5		Write A
6		Write B
7	Read B	
8	Write B	

A. This schedule is serializable and can occur in a scheme using 2PL protocol

B. This schedule is serializable but cannot occur in a scheme using 2PL protocol

C. This schedule is not serializable but can occur in a scheme using 2PL protocol

D. This schedule is not serializable and cannot occur in a scheme using 2PL protocol

unacademy

Question GATE-2003

Which of the following scenarios may lead to an irrecoverable error in a database system?

A. A transaction writes a data item after it is read by an uncommitted transaction
B. A transaction reads a data item after it is read by an uncommitted transaction
C. A transaction reads a data item after it is written by a committed transaction
D. A transaction reads a data item after it is written by an uncommitted transaction

Question GATE-2003

Consider three data items $D1, D2,$ and $D3,$ and the following execution schedule of transactions $T1, T2,$ and $T3.$ In the diagram, $R(D)$ and $W(D)$ denote the actions reading and writing the data item D respectively.

T1	T2	T3
	R(D3);	
	R(D2);	
	W(D2);	
		R(D2);
		R(D3);
R(D1);		
W(D1);		
		W(D2);
		W(D3);
	R(D1);	
R(D2);		
W(D2);		
	W(D1);	

Which of the following statements is correct?

A. The schedule is serializable as $T2; T3; T1$

B. The schedule is serializable as $T2; T1; T3$

C. The schedule is serializable as $T3; T2; T1$

D. The schedule is not serializable

unacademy

Question GATE-2004

Consider the following schedule S of transactions $T1$ and $T2$:

T1	T2
Read(A)	
A = A − 10	
	Read(A)
	Temp = 0.2*A
	Write(A)
	Read(B)
Write(A)	
Read(B)	
B = B + 10	
Write(B)	
	B = B + Temp
	Write(B)

Which of the following is TRUE about the schedule S ?

A. S is serializable only as $T1, T2$
B. S is serializable only as $T2, T1$
C. S is serializable both as $T1, T2$ and $T2, T1$
D. S is not serializable either as $T1, T2$ or as $T2, T1$

unacademy

Question GATE-2005

Amongst the ACID properties of a transaction, the 'Durability' property requires that the changes made to the database by a successful transaction persist

A. Except in case of an Operating System crash
B. Except in case of a Disk crash
C. Except in case of a power failure
D. Always, even if there is a failure of any kind

Question GATE-2005

A company maintains records of sales made by its salespersons and pays them commission based on each individual's total sales made in a year. This data is maintained in a table with following schema:

salesinfo = (salespersonid, totalsales, commission)

In a certain year, due to better business results, the company decides to further reward its salespersons by enhancing the commission paid to them as per the following formula:

If commission ≤ 50000, enhance it by 2%
If $50000 <$ commission ≤ 100000, enhance it by 4%
If commission > 100000, enhance it by 6%

The IT staff has written three different SQL scripts to calculate enhancement for each slab, each of these scripts is to run as a separate transaction as follows:

T1
```
Update salesinfo
Set commission = commission * 1.02
Where commission < = 50000;
```

T2
```
Update salesinfo
Set commission = commission * 1.04
Where commission > 50000 and commission is < = 100000;
```

T3
```
Update salesinfo
Set commission = commission * 1.06
Where commission > 100000;
```

Which of the following options of running these transactions will update the commission of all salespersons correctly

A. Execute T1 followed by T2 followed by T3
B. Execute T2, followed by T3; T1 running concurrently throughout
C. Execute T3 followed by T2; T1 running concurrently throughout
D. Execute T3 followed by T2 followed by T1

unacademy

Question GATE-2007

Consider the following schedules involving two transactions. Which one of the following statements is **TRUE**?

- $S_1 : r_1(X); r_1(Y); r_2(X); r_2(Y); w_2(Y); w_1(X)$
- $S_2 : r_1(X); r_2(X); r_2(Y); w_2(Y); r_1(Y); w_1(X)$

A. Both S_1 and S_2 are conflict serializable.

B. S_1 is conflict serializable and S_2 is not conflict serializable.

C. S_1 is not conflict serializable and S_2 is conflict serializable.

D. Both S_1 and S_2 are not conflict serializable.

unacademy

Question GATE-2007

Consider the following two transactions : T_1 and T2.

T_1 :
```
read (A);
read (B);
if A = 0 then B ← B + 1;
write (B);
```

T_2 :
```
read (B);
read (A);
if B ≠ 0 then A ← A - 1;
write (A);
```

Which of the following schemes, using shared and exclusive locks, satisfy the requirements for strict two phase locking for the above transactions?

A.

S1 :
```
lock S(A);
read (A);
lock S(B);
read (B);
if A = 0
then B ← B + 1;
write (B);
commit;
unlock (A);
unlock (B);
```

S2 :
```
lock S(B);
read (B);
lock S(A);
read (A);
if B ≠ 0
then A ← A - 1;
write (A);
commit;
unlock (B);
unlock (A);
```

B.

S1 :
```
lock X(A);
read (A);
lock X(B);
read (B);
if A = 0
then B ← B + 1;
write (B);
unlock (A);
commit;
unlock (B);
```

S2 :
```
lock X(B);
read (B);
lock X(A);
read (A);
if B ≠ 0
then A ← A - 1;
write (A);
unlock (A);
commit;
unlock (A);
```

C.

S1 :
```
lock S(A);
read (A);
lock X(B);
read (B);
if A = 0
then B ← B + 1;
write (B);
unlock (A);
commit;
unlock (B);
```

S2 :
```
lock S(B);
read (B);
lock X(A);
read (A);
if B ≠ 0
then A ← A - 1;
write (A);
unlock (B);
commit;
unlock (A);
```

D.

S1 :
```
lock S(A);
read (A);
lock X(B);
read (B);
if A = 0
then B ← B + 1;
write (B);
unlock (A);
unlock (B);
commit;
```

S2 :
```
lock S(B);
read (B);
lock X(A);
read (A);
if B ≠ 0
then A ← A - 1;
write (A);
unlock (A);
unlock (B);
commit;
```

unacademy

Question GATE-2008

Consider the following three schedules of transactions T1, T2 and T3. [Notation: In the following NYO represents the action Y (R for read, W for write) performed by transaction N on object O.]

$(S1)$	$2RA$	$2WA$	$3RC$	$2WB$	$3WA$	$3WC$	$1RA$	$1RB$	$1WA$	$1WB$
$(S2)$	$3RC$	$2RA$	$2WA$	$2WB$	$3WA$	$1RA$	$1RB$	$1WA$	$1WB$	$3WC$
$(S3)$	$2RA$	$3RC$	$3WA$	$2WA$	$2WB$	$3WC$	$1RA$	$1RB$	$1WA$	$1WB$

Which of the following statements is TRUE?

A. S1, S2 and S3 are all conflict equivalent to each other
B. No two of S1, S2 and S3 are conflict equivalent to each other
C. S2 is conflict equivalent to S3, but not to S1
D. S1 is conflict equivalent to S2, but not to S3

unacademy

Question GATE-2009

Consider two transactions T_1 and T_2, and four schedules S_1, S_2, S_3, S_4, of T_1 and T_2 as given below:

$T_1 : R_1[x]W_1[x]W_1[y]$

$T_2 : R_2[x]R_2[y]W_2[y]$

$S_1 : R_1[x]R_2[x]R_2[y]W_1[x]W_1[y]W_2[y]$

$S_2 : R_1[x]R_2[x]R_2[y]W_1[x]W_2[y]W_1[y]$

$S_3 : R_1[x]W_1[x]R_2[x]W_1[y]R_2[y]W_2[y]$

$S_4 : R_2[x]R_2[y]R_1[x]W_1[x]W_1[y]W_2[y]$

Which of the above schedules are conflict-serializable?

A. S_1 and S_2 B. S_2 and S_3 C. S_3 only D. S_4 only

unacademy

Question GATE-2010

Which of the following concurrency control protocols ensure both conflict serializability and freedom from deadlock?

I. 2-phase locking
II. Time-stamp ordering

A. I only B. II only C. Both I and II D. Neither I nor II

Question GATE-2010

Consider the following schedule for transactions $T1, T2$ and $T3$:

T1	T2	T3
Read(X)		
	Read(Y)	
		Read(Y)
	Write(Y)	
Write(X)		
		Write(X)
	Read(X)	
	Write(X)	

Which one of the schedules below is the correct serialization of the above?

A. $T1 \rightarrow T3 \rightarrow T2$
B. $T2 \rightarrow T1 \rightarrow T3$
C. $T2 \rightarrow T3 \rightarrow T1$
D. $T3 \rightarrow T1 \rightarrow T2$

unacademy

Question GATE-2012

Consider the following transactions with data items P and Q initialized to zero:

T_1	read (P); read (Q); if P = 0 then Q := Q + 1; write (Q)
T_2	read (Q); read (P); if Q = 0 then P := P + 1; write (P)

Any non-serial interleaving of **T1** and **T2** for concurrent execution leads to

A. a serializable schedule
B. a schedule that is not conflict serializable
C. a conflict serializable schedule
D. a schedule for which a precedence graph cannot be drawn

unacademy

Question GATE-2014

Consider the following four schedules due to three transactions (indicated by the subscript) using *read* and *write* on a data item x, denoted by $r(x)$ and $w(x)$ respectively. Which one of them is conflict serializable?

A. $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x);$
B. $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x);$
C. $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x);$
D. $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x);$

unacademy

Question GATE-2014

Consider the following schedule **S** of transactions $T1, T2, T3, T4$:

T1	T2	T3	T4
	Reads(X)		
		Writes(X)	
		Commit	
Writes(X)			
Commit			
	Writes(Y)		
	Reads(Z)		
	Commit		
			Reads(X)
			Reads(Y)
			Commit

Which one of the following statements is CORRECT?

A. **S** is conflict-serializable but not recoverable
B. **S** is not conflict-serializable but is recoverable
C. **S** is both conflict-serializable and recoverable
D. **S** is neither conflict-serializable not is it recoverable

unacademy

Question GATE-2014

Consider the transactions $T1, T2,$ and $T3$ and the schedules $S1$ and $S2$ given below.

$T1 : r1(X); r1(Z); w1(X); w1(Z)$

$T2 : r2(Y); r2(Z); w2(Z)$

$T3 : r3(Y); r3(X); w3(Y)$

$S1 : r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); r1(Z); w1(X); w1(Z)$

$S2 : r1(X); r3(Y); r2(Y); r3(X); r1(Z); r2(Z); w3(Y); w1(X); w2(Z); w1(Z)$

Which one of the following statements about the schedules is **TRUE**?

A. Only $S1$ is conflict-serializable.
B. Only $S2$ is conflict-serializable.
C. Both $S1$ and $S2$ are conflict-serializable.
D. Neither $S1$ nor $S2$ is conflict-serializable.

unacademy

Question GATE-2015

Consider the following transaction involving two bank accounts x and y.

```
read(x); x:=x-50; write (x); read(y); y:=y+50; write(y)
```

The constraint that the sum of the accounts x and y should remain constant is that of

A. Atomicity B. Consistency C. Isolation D. Durability

unacademy

Question GATE-2015

Consider a simple checkpointing protocol and the following set of operations in the log.

(start, T4); (write, T4, y, 2, 3); (start, T1); (commit, T4); (write, T1, z, 5, 7);

(checkpoint);

(start, T2); (write, T2, x, 1, 9); (commit, T2); (start, T3); (write, T3, z, 7, 2);

If a crash happens now and the system tries to recover using both undo and redo operations, what are the contents of the undo list and the redo list?

A. Undo: T3, T1; Redo: T2
B. Undo: T3, T1; Redo: T2, T4
C. Undo: none; Redo: T2, T4, T3, T1
D. Undo: T3, T1, T4; Redo: T2

unacademy

Question GATE-2015

Consider the partial Schedule S involving two transactions $T1$ and $T2$. Only the *read* and the *write* operations have been shown. The *read* operation on data item P is denoted by *read(P)* and *write* operation on data item P is denoted by *write(P)*.

Schedule S

Time Instance	Transaction ID	
	T1	**T2**
1	read(A)	
2	write(A)	
3		read(C)
4		write(C)
5		read(B)
6		write(B)
7		read(A)
8		commit
9	read(B)	

Suppose that the transaction $T1$ fails immediately after time instance 9. Which of the following statements is correct?

A. $T2$ must be aborted and then both $T1$ and $T2$ must be re-started to ensure transaction atomicity
B. Schedule S is non-recoverable and cannot ensure transaction atomicity
C. Only $T2$ must be aborted and then re-started to ensure transaction atomicity
D. Schedule S is recoverable and can ensure transaction atomicity and nothing else needs to be done

unacademy

Question GATE-2016

Which one of the following is NOT a part of the ACID properties of database transactions?

A. Atomicity B. Consistency C. Isolation D. Deadlock-freedom

unacademy

Question GATE-2016

Consider the following two phase locking protocol. Suppose a transaction T accesses (for read or write operations), a certain set of objects $\{O_1, \ldots . O_k\}$. This is done in the following manner:

Step 1 . T acquires exclusive locks to $O_1, \ldots . , O_k$ in increasing order of their addresses.

Step 2 . The required operations are performed .

Step 3 . All locks are released

This protocol will

A. guarantee serializability and deadlock-freedom
B. guarantee neither serializability nor deadlock-freedom
C. guarantee serializability but not deadlock-freedom
D. guarantee deadlock-freedom but not serializability.

unacademy

Question GATE-2016

Suppose a database schedule S involves transactions $T_1, \ldots\ldots, T_n$. Construct the precedence graph of S with vertices representing the transactions and edges representing the conflicts.If S is serializable, which one of the following orderings of the vertices of the precedence graph is guaranteed to yield a serial schedule?

A. Topological order
B. Depth-first order
C. Breadth- first order
D. Ascending order of the transaction indices

unacademy

Question GATE-2016

Consider the following database schedule with two transactions T_1 and T_2.

$S = r_2(X); r_1(X); r_2(Y); w_1(X); r_1(Y); w_2(X); a_1; a_2$

Where $r_i(Z)$ denotes a read operation by transaction T_i on a variable Z, $w_i(Z)$ denotes a write operation by T_i on a variable Z and a_i denotes an abort by transaction T_i.

Which one of the following statements about the above schedule is **TRUE**?

A. S is non-recoverable.
B. S is recoverable, but has a cascading abort.
C. S does not have a cascading abort.
D. S is strict.

unacademy

Question GATE-2019

Consider the following two statements about database transaction schedules:

I. Strict two-phase locking protocol generates conflict serializable schedules that are also recoverable.
II. Timestamp-ordering concurrency control protocol with Thomas' Write Rule can generate view serializable schedules that are not conflict serializable

Which of the above statements is/are TRUE?

A. I only B. II only C. Both I and II D. Neither I nor II

unacademy

Happy Learning.!