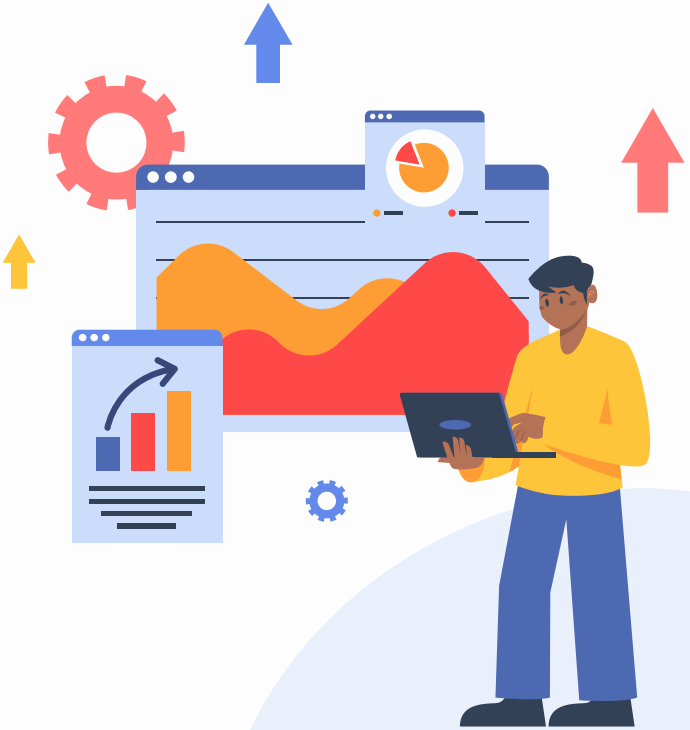
Trees

By Aditi Mam



June 2014 Paper II

The upper bound and lower bound for the number of leaves in a B-Tree of degree K with height h is given by:

A. K^h and $2\lceil \frac{k}{2} \rceil^{h-1}$

B. $K * h$ and $2\lfloor \frac{k}{2} \rfloor^{h-1}$

C. K^h and $2\lfloor \frac{k}{2} \rfloor^{h-1}$

D. $K * h$ and $2\lceil \frac{k}{2} \rceil^{h-1}$

June 2014 Paper II

The upper bound and lower bound for the number of leaves in a B-Tree of degree K with height h is given by:

A. K^h and $2\lceil \frac{k}{2} \rceil^{h-1}$

B. $K * h$ and $2\lfloor \frac{k}{2} \rfloor^{h-1}$

C. K^h and $2\lfloor \frac{k}{2} \rfloor^{h-1}$

D. $K * h$ and $2\lceil \frac{k}{2} \rceil^{h-1}$

Answer: A

Paper III June 2014

Suppose that the splits at every level of quicksort are in the proportion $(1 - \alpha)$ to a, where $0 < \alpha \leq \frac{1}{2}$ is a constant. The minimum depth of a leaf in the recursion tree is approximately given by

(A) $-\frac{lgn}{lg(1-\alpha)}$

(B) $-\frac{lg\,(1-\alpha)}{lgn}$

(C) $-\frac{lgn}{lg\alpha}$

(D) $-\frac{lg\alpha}{lgn}$

Answer: C

if we split with α, we will get minimum depth. suppose α=1/2 means pivot is the middle element. with such case recursion tree will be with minimum depth. other cases it will have maximum.

1. nα^k=1
k=-logn/logα

2. n(1-α)^k=1
k=-logn/log(1-α)

Paper II December 2014

A full binary tree with n leaves contains

(A) n nodes
(B) $\log_2$ n nodes
(C) 2n –1 nodes
(D) 2^n nodes

(C) 2n –1 nodes

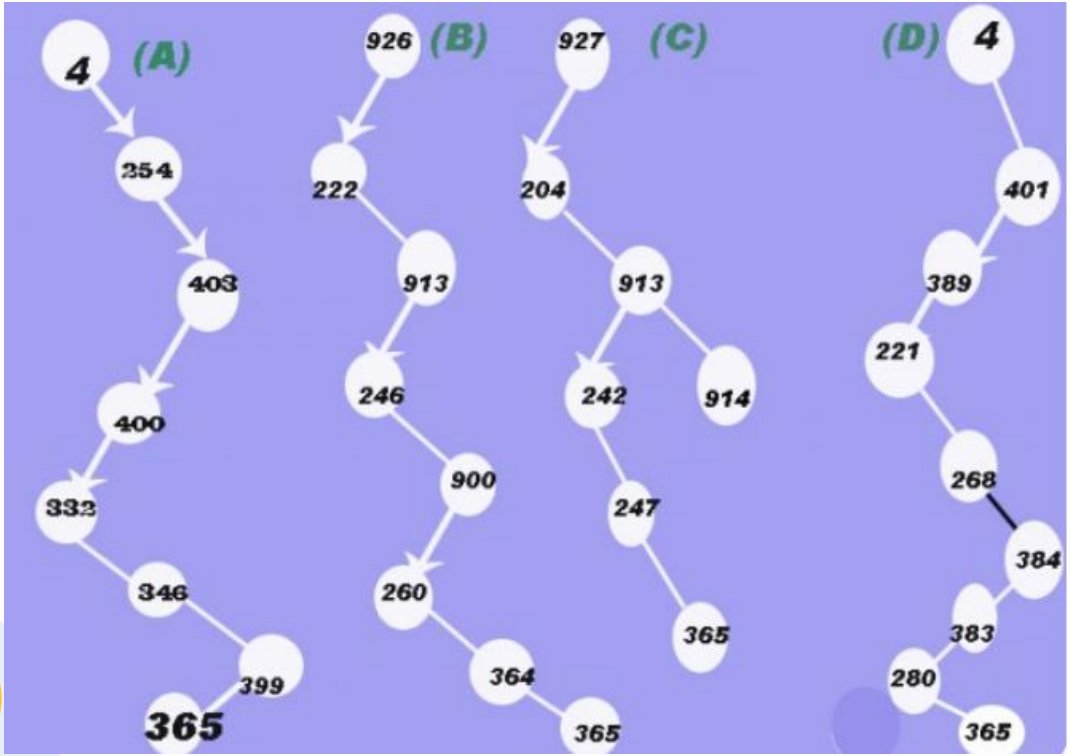
A full binary tree with n leaves contains 2n-1 nodes.

Paper III December 2014

Suppose that we have numbers between 1 and 1000 in a binary search tree and we want to search for the number 365. Which of the following sequences could not be the sequence of nodes examined ?

(A) 4, 254, 403, 400, 332, 346, 399, 365
(B) 926, 222, 913, 246, 900, 260, 364, 365
(C) 927, 204,913, 242, 914, 247, 365
(D) 4, 401, 389, 221, 268, 384, 383, 280, 365

(C) 927, 204,913, 242, 914, 247, 365

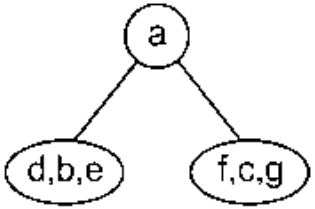


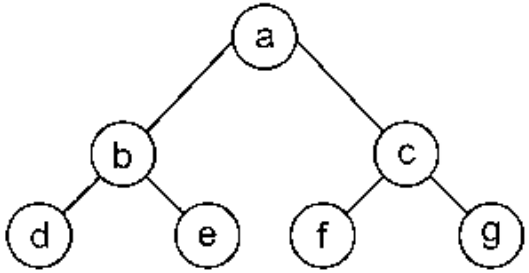
Paper II June 2015

The inorder and preorder Traversal of binary Tree are dbeafcg and abdecfg respectively. The post-order Traversal is .....

(A) dbefacg
(B) debfagc
(C) Dbefcga
(D) debfgca

(D) debfgca





postorder traversal : d e b f g c a

Paper II June 2015

Level order Traversal of a rooted Tree can be done by starting from root and performing:

(A) Breadth First Search
(B) Depth first search
(C) Root search
(D) Deep search

(A) Breadth First Search

Level order traversal of rooted tree can be done by starting from root and visiting all node at k level before visiting any node at k + 1 level, which is nothing but breadth first search.

Paper III June 2015

The number of nodes of height h in any n - element heap is .....

1. $\text{ceil}[\frac{n}{2^{h+1}}]$
2. $\frac{n}{2^{h-1}}$
3. $\frac{n}{2^{h}}$
4. $\frac{n-1}{2^{h-1}}$

Answer: A

Consider a binary heap of 7 nodes, i.e., n=7.

It would be a complete binary tree of height 2.

Number of nodes at height 0, i.e., at the lowest level = 4 = $\lceil \frac{7}{2^{0+1}} \rceil$.

Number of nodes at height 1 = 2 = $\lceil \frac{7}{2^{1+1}} \rceil$.

Number of nodes at height 2, i.e., at root level 1 = $\lceil \frac{7}{2^{2+1}} \rceil$

So, the correct option is option 1) ceil$\lceil \frac{n}{2^{h+1}} \rceil$

Paper II December 2015

The number of disk pages access in B-tree search, where h is height, n is the number of keys, and t is the minimum degree, is:

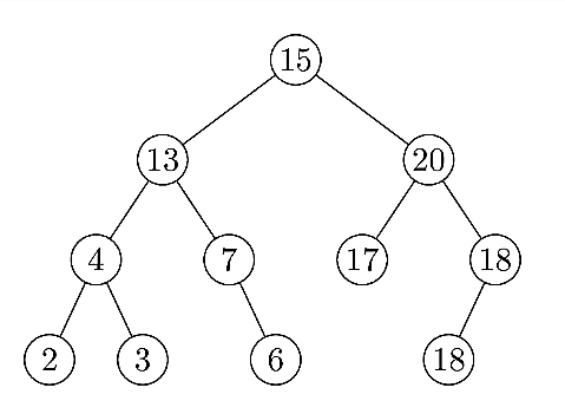
(A) $\theta(\log_n h*t)$
(B) $\theta(\log_t n*h)$
(C) $\theta(\log_h n)$
(D) $\theta(\log_t n)$

(D) $\theta(\log_t n)$

The number of disk pages accessed by B-TREE-SEARCH is $\Theta(h) = \Theta(\log_t n)$, where h is the height of the B-tree and n is the number of keys in the B-tree.

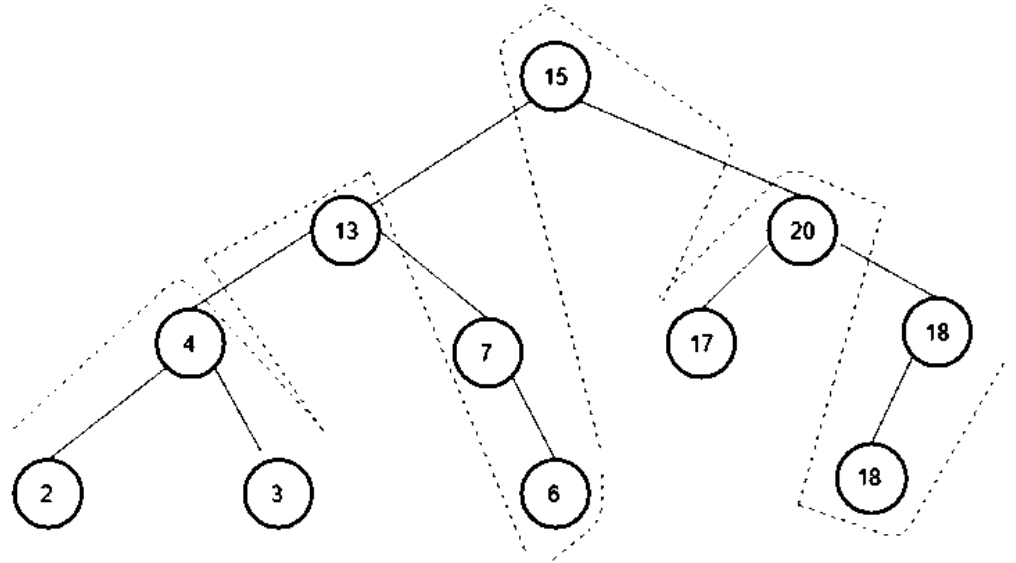
Paper II December 2015

The inorder traversal of the following tree is:



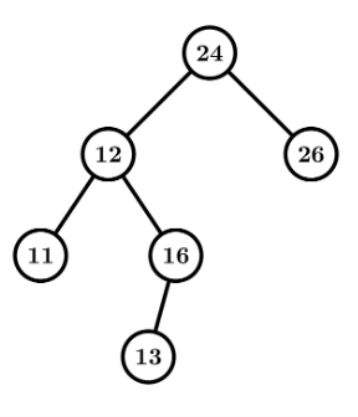
(A) 2 3 4 6 7 13 15 17 18 18 20
(B) 20 18 18 17 15 13 7 6 4 3 2
(C) 15 13 20 4 7 17 18 2 3 6 18
(D) 2 4 3 13 7 6 15 17 20 18 18

(D) 2 4 3 13 7 6 15 17 20 18 18



Paper II July 2016

Consider the following binary search tree:

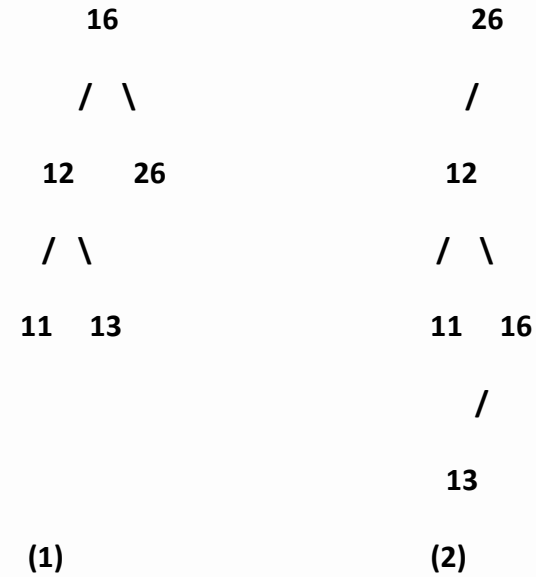


If we remove the root node, which of the node from the left subtree will be the new root?

(A) 11 **(B) 12**
(C) 13 **(D) 16**

(D) 16

Here root node 24 has children two. After deleting 24 we can replace root element by inorder successor or inorder predecessor of the node.



Either element 16 of left sub-tree or element 26 of right sub-tree will be next root element.

Paper II July 2016

Suppose you are given a binary tree with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be

(A) n/2 – 1
(B) n/2 + 1
(C) (n-1)/2
(D) (n+1)/2

(C) (n-1)/2

If it is left or right biased than gives the maximum height...

Maximum height is(n-1/ 2) taking root at height 0.

Paper III July 2016

The number of different binary trees with 6 nodes is .....

(A) 6
(B) 42
(C) 132
(D) 256

(C) 132

(2n)!/(n+1)!n! different binary trees are possible with n nodes.

Given n=6

Number of trees = 12!/(7!x 6!) =132

Paper III August 2016 (Re-test)

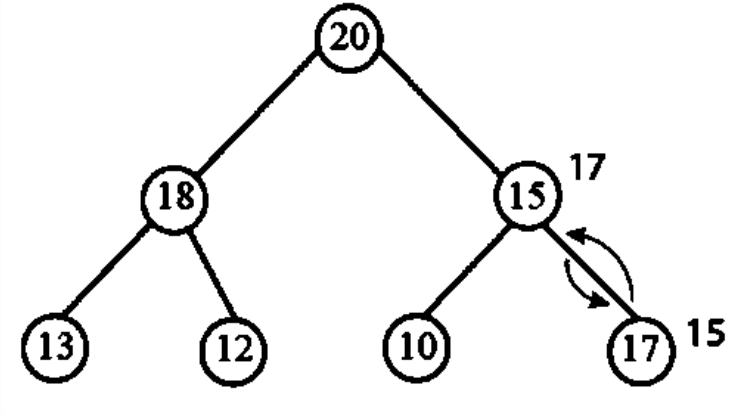
A priority queue is implemented as a max-heap. Initially, it has five elements. The level-order traversal of the heap is as follows:

20, 18, 15, 13, 12

Two new elements ‘10’ and ‘17’ are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the element is:

(A) 20, 18, 17, 15, 13, 12, 10
(B) 20, 18, 17, 12, 13, 10, 15
(C) 20, 18, 17, 10, 12, 13, 15
(D) 20, 18, 17, 13, 12, 10, 15

(D) 20, 18, 17, 13, 12, 10, 15



Paper III July 2016

Which one of the following array represents a binary max-heap?

(A) [26, 13, 17, 14, 11, 9, 15]
(B) [26, 15, 14, 17, 11, 9, 13]
(C) [26, 15, 17, 14, 11, 9, 13]
(D) [26, 15, 13, 14, 11, 9, 17]

(C) [26, 15, 17, 14, 11, 9, 13]

For max heap we will compare parent node(i) with its left-child(2 * i) and right-child(2 * i + 1):
In first option node(2) < node(5) which is violating the max-heap property.
In second option node node(2) < node(5) which is violating the max-heap property.
In third option there is no violation.
In fourth option node(3) < node(7) which is violating the max-heap property.
So, option (C) is correct.

Paper III July 2016

Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 364. Which of the following sequences could not be the sequence of nodes examined?

(A) 925, 221, 912, 245, 899, 259, 363, 364
(B) 3, 400, 388, 220, 267, 383, 382, 279, 364
(C) 926, 203, 912, 241, 913, 246, 364
(D) 3, 253, 402, 399, 331, 345, 398, 364

(C) 926, 203, 912, 241, 913, 246, 364

913 can not be searched after 912 after 912 any number must be less than 912 since 912 >364 so we will move towards 912 to 1

Paper II August 2016 (Re-test)

Consider an undirected graph G where self-loops are not allowed. The vertex set of G is {(i, j) | 1 ≤ i ≤ 12, 1 ≤ j ≤ 12}. There is an edge between (a, b) and (c, d) if |a – c| ≤ 1 or |b–d| ≤ 1. The number of edges in this graph is

(A) 726 (B) 796

(C) 506 (D) 616

(C) 506

If you think of a 12×12 grid (like a chess board of size 12×12), then each each point (i, j), which is in i^{th} row and j^{th} column, is a vertex (i, j).

Now we are allowed to connect only those points which are atmost 1 distance apart (in both horizontal and vertical direction). So we will connect only horizontal neighbours, vertical neighbours, and diagonal neighbours.

So horizontal edges on each row are 11 i.e. $11 \times 12 = 132$ horizontal edges. Similarly we have 132 vertical edges.

To count diagonal edges, think of 1×1 square boxes in which diagonals meet each other. There are 11×11 such square boxes, and each box contains 2 diagonals, so total diagonals $= 242$.

So total edges $= 132 + 132 + 242 = 506.$

Paper III August 2016 (Re-test)

Consider a weighted complete graph G on the vertex set {v1, v2, …. vn} such that the weight of the edge (vi, vj) is 4 |i – j|. The weight of minimum cost spanning tree of G is:

(A) 4n2 **(B) n**

(C) 4n – 4 **(D) 2n – 2**

Paper III August 2016 (Re-test)

Consider a weighted complete graph G on the vertex set {v1, v2, …. vn} such that the weight of the edge (vi, vj) is 4 |i – j|. The weight of minimum cost spanning tree of G is:

(A) 4n2 **(B) n**

(C) 4n – 4 **(D) 2n – 2**

Paper II January 2017

Which of the following statements is false?
(A) Optimal binary search tree construction can be performed efficiently using dynamic programming.
(B) Breadth-first search cannot be used to find connected components of a graph.
(C) Given the prefix and postfix walks of a binary tree, the tree cannot be re-constructed uniquely.
(D) Depth-first-search can be used to find the components of a graph.

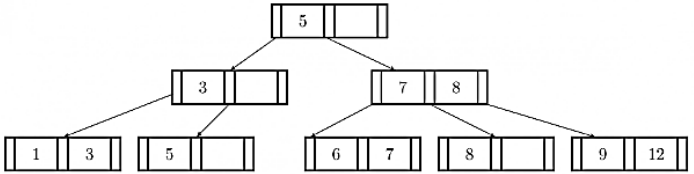
(B) Breadth-first search cannot be used to find connected components of a graph.

BFS and DFS both algorithm are used to find connected component.

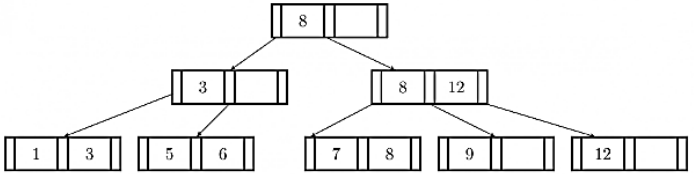
Paper III January 2017

If following sequence of keys are inserted in a B+ tree with K(=3) pointers:
8, 5, 1, 7, 3, 12, 9, 6
Which of the following shall be correct B+ tree?

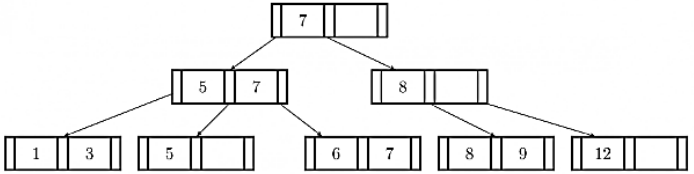
A.



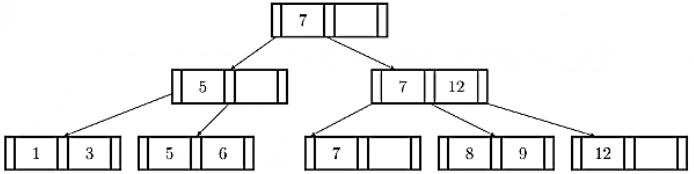
B.



C.

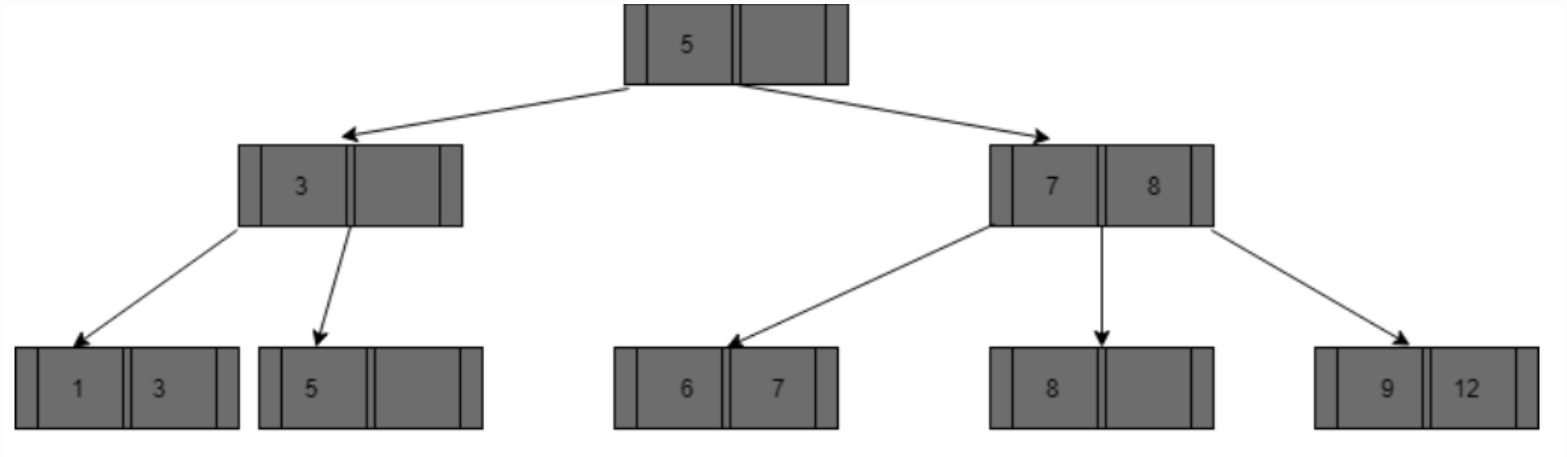


D.



Answer: 1

B+ is either left biased (<=) or it is right biased (>=), all trees except (1) is violating the biasing property.



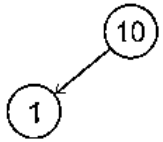
Paper II November 2017

The following numbers are inserted into an empty binary search tree in the given order:
10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree?
(1) 3
(2) 4
(3) 5
(4) 6

(1) 3

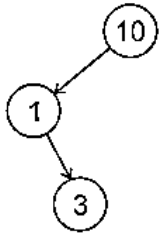
Step 1: First 10 comes and now that is the **Root** node.

Step 2: Now 1 came and 1 < 10 then insert Node 1 to the **Left** of Node 10.



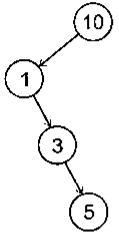
Step 3: Now 3 came and 3 < 10 go to the **Left** of

Node 10 and check 3 > 1 then insert Node 3 to the **Right** of Node 1.

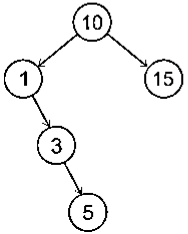


Step 4: Now 5 came and 5 < 10 go to the **Left** of

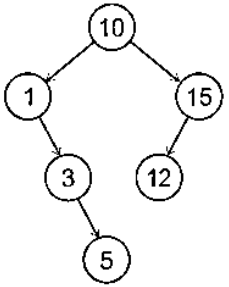
Node 10 and check 5 > 1 go to the **Right** of Node 1 then check 5 > 3 then insert Node 5 to the **Right** of Node 3.



Step 5: Now 15 came and 15 > 10 then insert Node 15 to the **Right** of Node 10.

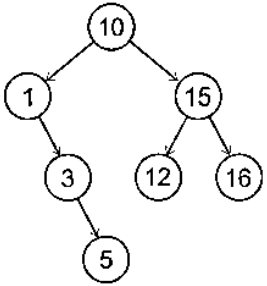


Step 6: Now 12 came and 12 > 10 go to the **Right** of Node 10 and check 15 > 12 then insert Node 12 to the **Left** of Node 15.



Step 7: Now 16 came and 16 > 10 go to the **Right** of

10 and check 16 > 15 then insert 16 to the **Right** of Node 15.



After step 7, we can count the height of the tree as 3.

Paper II November 2017

Let G be an undirected connected graph with distinct edge weight. Let Emax be the edge with maximum weight and Emin the edge with minimum weight. Which of the following statements is false?

(1) Every minimum spanning tree of G must contain Emin.

(2) If Emax is in minimum spanning tree, then its removal must disconnect G.

(3) No minimum spanning tree contains Emax.

(4) G has a unique minimum spanning tree.

(3) No minimum spanning tree contains Emax.

Paper II November 2017

Postorder traversal of a given binary search tree T produces following sequence of keys:
3, 5, 7, 9, 4, 17, 16, 20, 18, 15, 14
Which one of the following sequences of keys can be the result of an in-order traversal of the tree T?
(1) 3, 4, 5, 7, 9, 14, 20, 18, 17, 16, 15
(2) 20, 18, 17, 16, 15, 14, 3, 4, 5, 7, 9
(3) 20, 18, 17, 16, 15, 14, 9, 7, 5, 4, 3
(4) 3, 4, 5, 7, 9, 14, 15, 16, 17, 18, 20

(4) 3, 4, 5, 7, 9, 14, 15, 16, 17, 18, 20

The in-order traversal of the Binary Search Tree(BST) always gives elements in ascending or increasing order. So, in-order traversal of the sequence of keys given in post-order traversal is sorted ascending order of that keys. Hence, the sequence of keys in an in-order traversal of BST are : 3, 4, 5, 7, 9, 14, 15, 16, 17, 18, 20

Paper II November 2017

Consider an array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array (i< = n), the index of the parent is:

(1) floor ((i +1)/2)

(2) ceiling ((i + 1)/2)

(3) floor (i/2)

(4) ceiling (i/2)

(3) floor (i/2)

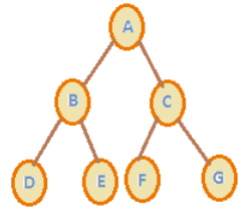
Binary heap elements can be represented using an array with index value i (i < = n), Parent node of element will be at index: floor (i / 2) Left Child node will be at index: 2 * i Right child node will be at index: 2 * i + 1 Hence, the index of the parent is floor (i / 2).

Paper III November 2017

Consider a full binary tree with n internal nodes, internal path length i, and external path length e. The internal path length of a full binary tree is the sum, taken over all nodes of the tree, of the depth of each node. Similarly, the external path length is the sum, taken over all leaves of the tree, of the depth of each leaf.
Which of the following is correct for the full binary tree?
(1) e = i + n
(2) e = i + 2n
(3) e = 2i + n
(4) e = 2n + i

(2) e = i + 2n

Consider the following full binary tree where number of internal nodes n = 3



Internal path length i = d(A) + d(B) + d(C) = 0 + 1 + 1 = 2
External path length e = d(D) + d(E) + d(F) + d(G) = 2 + 2 + 2 + 2 = 8
We can see that e = i + 2n = 2 + 2(3) = 8
You can verify this is true for any full binary tree.
e = i + 2n is the answer.

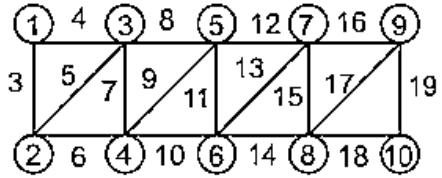
Paper III November 2017

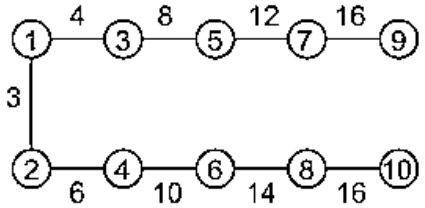
An undirected graph G (V, E) contains n (n > 2) nodes named v1, v2,...,vn. Two nodes vi and vj are connected if and only if $0 < |i - j| \leq 2$. Each edge (vi, vj) is assigned a weight i+j.
The cost of the minimum spanning tree of such a graph with 10 nodes is :
(1) 88
(2) 91
(3) 49
(4) 21

(2) 91

1. Two nodes v_i, and v_j are connected if and only if $0 < |i - j| \leq 2$
2. Each edge (v_i, v_j) is assigned a weight $i + j$.

Graph from the given conditions





$3 + 4 + 8 + 8 + 12 + 16 + 18 = 91$

Therefore the minimum cost is 91

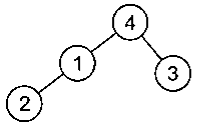
Paper II July 2018

Consider the array A=<4, 1, 3, 2, 16, 9, 10, 14, 8, 7>. After building heap from the array A, the depth of the heap and the right child of max-heap are ..... and .....
respectively. (Root is at level 0).
(1) 3, 14
(2) 3, 10
(3) 4, 14
(4) 4, 10

(2) 3, 10

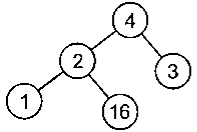
A = <4, 1, 3, 2, 16, 9, 10, 14, 8, 7>.

STEP 1



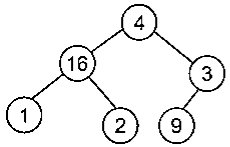
Now, As 2 is greater than 1, there is need of rearrangement of nodes.

STEP 2:

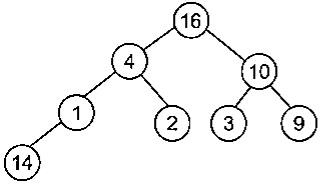


Again exchange 16 with 2. Insert remaining nodes as per the property of max- heap.

STEP 3:

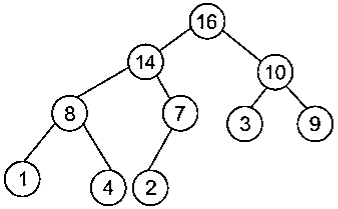


STEP 4:



STEP 5:

After inserting all the nodes in this way, final max - heap is:



As, root is at level 0, So, there are total 3 levels in this max - heap. Also, right child of root node is 10.

Paper II July 2018

A binary search tree in which every non-leaf node has non-empty left and right subtrees is called a strictly binary tree. Such a tree with 19 leaves:
(1) cannot have more than 37 nodes
(2) has exactly 37 nodes
(3) has exactly 35 nodes
(4) cannot have more than 35 nodes

(2) has exactly 37 nodes

L = I (n - 1) + 1
where,
L = number of leaf nodes
I = number of internal nodes
n = n - ary tree

L = 19 n = 2 (as given tree is strictly binary tree)
19 = I (2 - 1) + 1
I = 18
Number of internal nodes = 18
Number of leaf nodes = 19
Total number of nodes = 18 + 19 = 37

Paper II July 2018

A 5-ary tree is tree in which every internal node has exactly 5 children. The number of left nodes in such a tree with 8 internal nodes will be:
(1) 30
(2) 33
(3) 41
(4) 125

Paper II July 2018

A 5-ary tree is tree in which every internal node has exactly 5 children. The number of left nodes in such a tree with 8 internal nodes will be:

(1) 30

(2) 33

(3) 41

(4) 125

Paper II December 2018

The elements 42,25,30,40,22,35,26 are inserted one by one in the given order into a max-heap. The resultant max-heap is sorted in an array implementation as

(1) <42,40,35,25,22,30,26>
(2) <42,35,40,22,25,30,26>
(3) <42,40,35,25,22,26,30>
(4) <42,35,40,22,25,26,30>

(1) <42,40,35,25,22,30,26>

After inserting each element, we will apply MAX-Heapify operation to get the MAX-Heap.

Insert: 42, 25 and 30

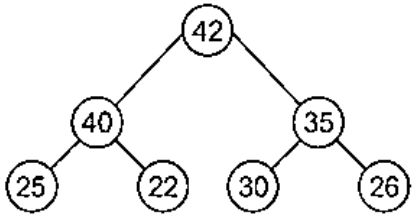
Insert: 40 apply MAX-Heapify

New order: 42, 40, 30, 25, 22

Insert: 35 apply MAX-Heapify

New order: 42, 40, 35, 25, 22, 30 and 26

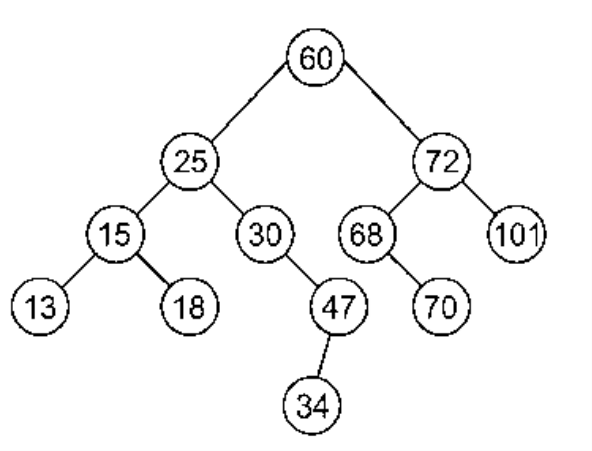
Diagram: resultant MAX_Heap



Paper II December 2018

A binary search tree is constructed by inserting the following numbers in order:
60, 25, 72, 15, 30, 68, 101, 13, 18, 47, 70, 34
The number of nodes in the left subtree is
(1) 5
(2) 6
(3) 7
(4) 3

(3) 7



Paper II December 2018

In a ternary tree, the number of internal nodes of degree 1, 2, and 3 is 4, 3, and 3 respectively. The number of leaf nodes in the ternary tree is

(1) 9
(2) 10
(3) 11
(4) 12

(2) 10

Number of leaf nodes = sum of degree of all internal nodes – total internal nodes + 1

Total number of internal nodes = 4+ 3 + 3 = 10
Sum of degree of all internal nodes = (4 × 1) + (3 × 2) + (3 × 3) = 4 + 6 + 9 = 19
Number of leaf nodes = sum of degree of all internal nodes – total internal nodes + 1 = 19 – 10 + 1 = 10

Paper II November 2020

A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L=41, and I=10. What is the value of n?

(1) 3
(2) 4
(3) 5
(4) 6

(3) 5

Given that leaves L= 41, internal nodes I=10
L=(n-1)I+1
41=10(n-1)+1
10n=50
n=5

Paper II October 2022

The number of nodes of height h in any n-element heap is at most

a) $n/2^{h+1}$
b) $n/2^{h-1}$
c) $n/2^{h}$
d) $n-1/2^{h-1}$

a) $n/2^{h+1}$

Consider a binary heap of 7 nodes, i.e., n=7.
It would be a complete binary tree of height 2.
Number of nodes at height 0, i.e., at the lowest level = 4 = ceil $7/2^{0+1}$
Number of nodes at height 1 = 2 = ceil $7/2^{1+1}$
Number of nodes at height 2, i.e., at root level 1 = ceil $7/2^{2+1}$

So, the correct option is option 1) ceil $n/2^{h+1}$

Paper II October 2022

Consider a B-tree of height h. minimum degree t >= 2 that contains any n key, where n >= 1 Which of the following is correct?

a) $h >= \log_t((n + 1)/2)$
b) $h <= \log_t((n + 1)/2)$
c) $h >= \log_t((n - 1)/2)$
d) $h <= \log_t((n - 1)/2)$

b) $h <= \log_t ((n + 1)/2)$

n the case of a B-tree with a minimum number of keys, there is one key in the root, 2 nodes at depth 1 , $2t^i$ -1 nodes at depth i .
Let h be the height of the tree and n be the number of nodes. Then n>=1+(t-1) submission up h down i=1 $2t^i$ -1

which works out to $t^h <= (n+1) / 2$. So we take the logt of both sides.
$h <= \log_t (n+1)/2$

Hence only $h <= \log_t ((n + 1)/2)$ is correct.

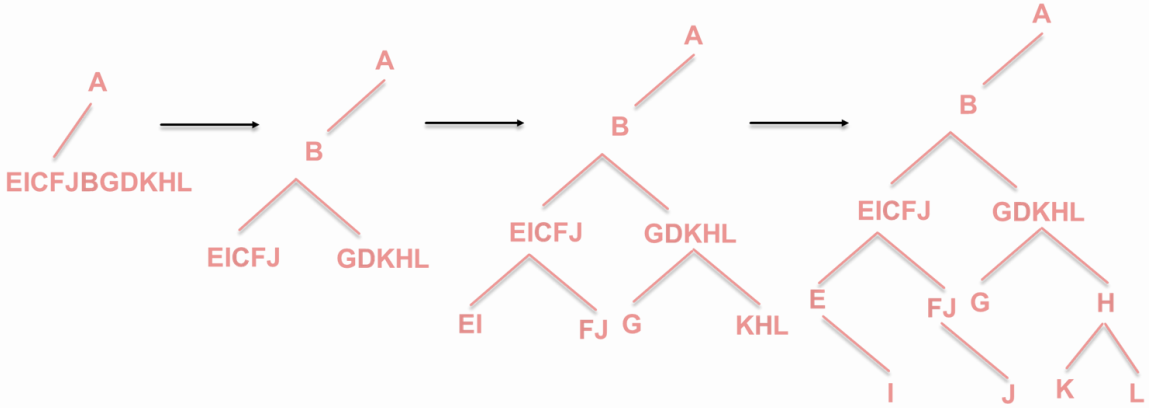
Paper II October 2022
Consider the traversal of a tree

Preorder- ABCEIFJDGHKL
Inorder- EICFJBGDKHLA

Which of the following is correct post order traversal?

a) **EIFJCKGLHDBA**
b) **FCGKLHDBUAE**
c) **FCGKLHDBAEIJ**
d) **IEJFCGKLHDBA**

D) IEJFCGKLHDBA



Postorder : IEJFCGKLHDBA

Paper II October 2022

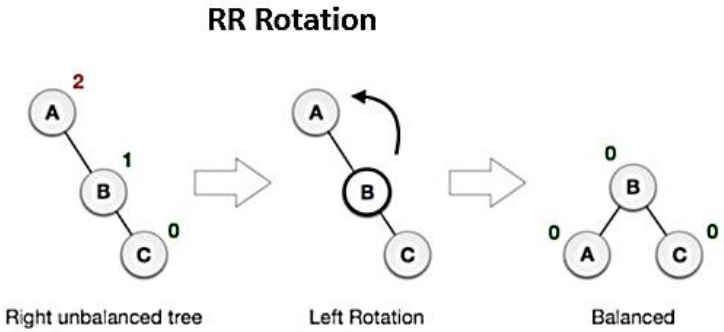
How many rotations are required during the construction of an AVL tree if the following elements are to be added in the given sequence?

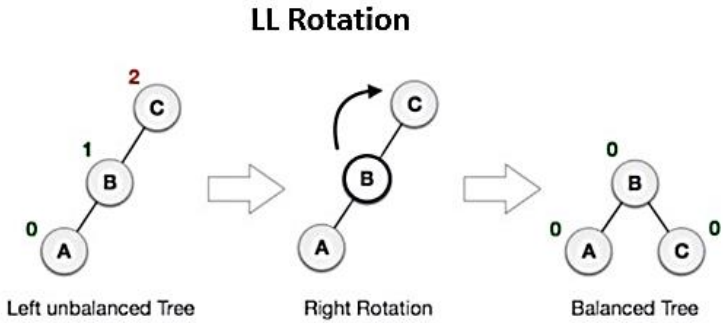
35, 50, 40, 25, 30, 60, 78, 20, 28

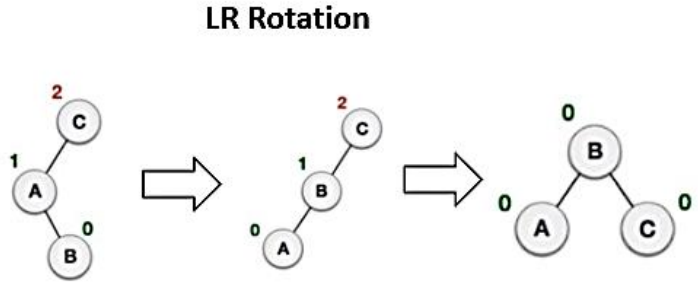
a) 2 left rotations, 2 right rotations
b) 2 left rotations, 3 right rotations
c) 3 left rotations, 2 right rotations
d) 3 left rotations, 1 right rotations

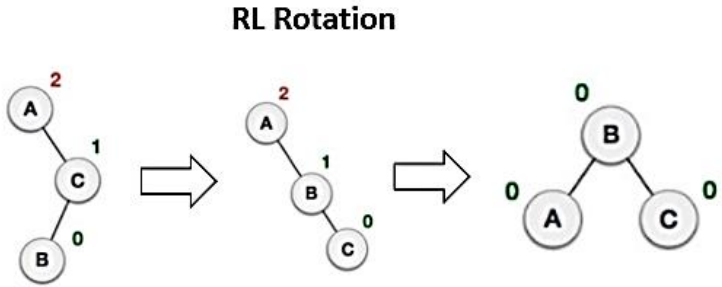
C) 3 left rotations, 2 right rotations

Rotations in AVL

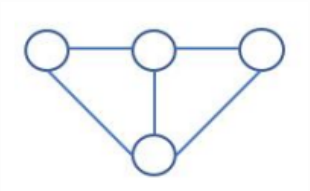








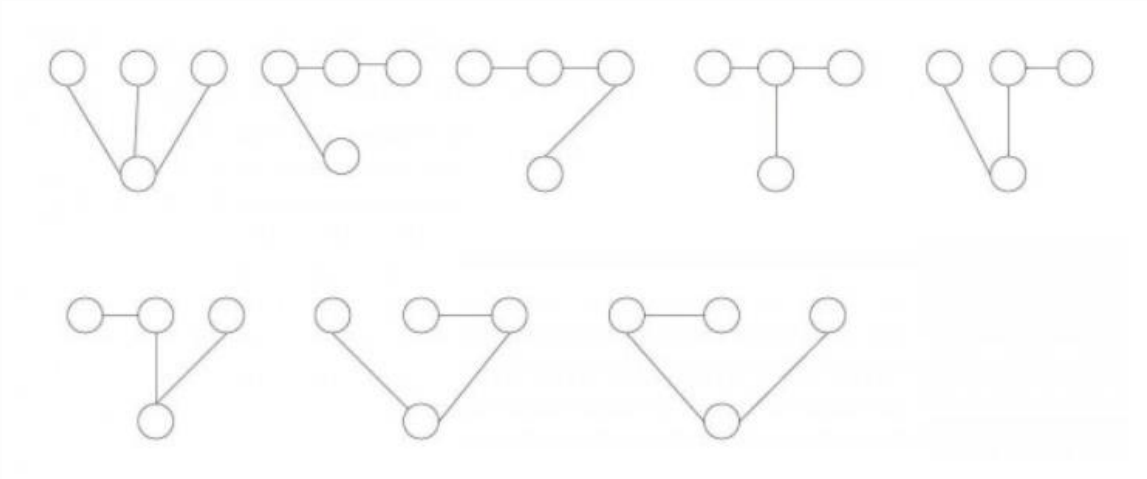
Paper II March 2023
Consider the Graph below:



How many spanning trees can be found?

1. **10**
2. **5**
3. **9**
4. **8**

4. 8



Paper II June 2023

Consider the following statements about heap sort algorithm:

A. **The MAX-HEAPIFY procedure which runs in O lg (n) time, is the key to maintaining the max heap property.**
B. **The BUILD-MAX-HEAP procedure, which runs in O lg (n) time, produces max-heap from an unordered input array.**
C. **The MAX-HEAP-INSERT, which rims in O (lg n) time, implements the insertion operation.**
D. **The HEAP-INCREASE-KEY procedure runs in O(n lg n) time, to set the key of new node of its correct value.**

Choose the correct answer from the options given below

1. A, B only
2. A, C only
3. B, D only
4. A, B, C, D

2. A, C only

- The MAX-HEAPIFY procedure, which runs in $O(\lg n)$ time, is the key to maintaining the max-heap property.
- The BUILD-MAX-HEAP procedure, which runs in linear time, produces a max-heap from an unordered input array.
- The HEAPSORT procedure, which runs in $O(n \lg n)$ time, sorts an array in place.
- The MAX-HEAP-INSERT, HEAP-EXTRACT-MAX, HEAP-INCREASE-KEY, and HEAP-MAXIMUM procedures, which run in $O(\lg n)$ time, allow the heap data structure to be used as a priority queue.

Paper II June 2023

Given below are two statements: one is labelled as Assertion A and the other is labelled as Reason R.

Assertion A: The AVL trees are more balanced as compared to Red Black trees, but they may cause more rotations during insertion and deletion

Reason R: A Red Black tree with n nodes has height that is greater than $2 \log_2 (n+1)$ and the AVL tree with n nodes has height less than $\log_\Phi \left(\sqrt{5}(n+2)\right) - 2$ (where Φ is golden ratio)

In the light of the above statements, choose the correct answer from the options given below.

1. Both A and R are correct and R is the correct explanation of A
2. Both A and R are correct and R is NOT the correct explanation of A
3. A is true but R is false
4. A is false but R is true

3.

Assertion A: "AVL trees are more balanced as compared to Red Black trees, but they may cause more rotations during insertion and deletion." This assertion is correct.

AVL trees are more strictly balanced than Red-Black trees, meaning that the height difference between the left and right subtrees of any node (called the balance factor) is at most 1 in AVL trees, whereas it can be up to 2 in Red-Black trees. However, this strict balancing in AVL trees can lead to more rotations during insertion and deletion operations compared to Red-Black trees.

Reason R: "A Red-Black tree with n nodes has a height that is greater than $2 \log_2 (n + 1)$, and the AVL tree with n nodes has a height less than $\log\Phi (\sqrt{5} (n+2)) - 2$ (where Φ is the golden ratio)." This reason is incorrect.

The correct upper bound for the height of a Red-Black tree with n nodes is $2 \log_2 (n + 1)$, not greater than. The statement for AVL trees is also incorrect; the correct upper bound for the height of an AVL tree with n nodes is approximately $1.44 * \log_2(n+2) - 1.329$.

Paper II June 2023

The total cost of retrieving records in sorted order using an unclustered B+ tree is

(P- Average number of records per data page

N- Data pages

F- Ratio of the size of a data entry to the size of a data record)

1. (F*N) +P
2. (F+P) * N
3. F*N*P
4. F+P/N

2.

When using an un-clustered B+ tree to retrieve records in sorted order, the total cost varies depending on the parameters you've provided: the average number of records per page (P), the number of data pages (N), and the ratio of the size of a data entry to the size of a data record (F).

We can calculate the cost as follows: We need to access each data page once N. However, for each data entry, because of the un-clustered nature of the B+ Tree, there might be repeated access to a data page.

So, the cost to retrieve the data entries is F*N. Secondary, we also need to consider the average number of records in a data page. When we retrieve the data, we have to consider this factor.

As we're retrieving records in sorted order, we need to look at each record at least once, so average number of records per page (P) comes into the play.

So from the given options, it seems like the closest one is: (F + P) * N which represents the cost of retrieving records in sorted order from an un-clustered B+ tree. It combines the cost of retrieving each data entries (F*N) and retrieving the ,records (P*N).

Paper II December 2023
Which of the following is/are NOT CORRECT statement?

(A) The first record in each block of the data file is known as actor record.
(B) Dense index has index entries for every search key value in the data file.
(C) Searching is harder in the B tree than B-tree as the all external nodes linked to each other.
(D) In extendible hashing the size of directory is just an array of 2^{d-1}, where d is global depth.

Choose the correct answer from the options given below:
(1) (A), (B) and (C) Only
(2) (A), (C) and (D) Only
(3) (A), (B) and (D) Only
(4) (A), (B), (C) and (D) Only

(2) (A), (C) and (D) Only

(A) The first record in each block of the data file is known as actor record. The first record in each block of the data file is not known as an actor record, this seems to be an incorrect usage of terminology in the context of databases.

(C) Searching is harder in the B+ tree than B- tree as the all external nodes linked to each other. This statement is incorrect. Searching in a B+ tree is not harder than in a B- tree. In fact, B+ trees are often preferred over B- trees for database indexing because they allow for efficient traversal through all keys (all keys are stored in the leaves in a sorted manner and leaves are linked for easy traversal).

(D) In extendible hashing the size of directory is just an array of 2^{d-1}, where d is global depth. In extendible hashing, the size of the directory is not an array of 2^{d-1}. It is actually an array of 2^{d}, where 'd' stands for the global depth.

Paper II December 2023

2 - 3 - 4 trees are B- trees of order 4. They are isometric of _____ trees.

(1) AVL
(2) AA
(3) 2-3
(4) Red-Black

(4) Red-Black

2-3-4 trees are isometric of Red-Black trees. It means that, for every 2-3-4 tree, there exists a Red-Black tree with data elements in the same order.