**QUESTIONS & ANSWERS**

**QUES:-** What is the MAXIMUM number of KEYS in a B-Tree of order m of height h?

**Explanation:-**

1 node m-1 keys at level 1

m nodes m \* (m-1) keys at level 2

m\*\*2 nodes m \* m \* (m-1) keys at level 3

...

m\*\*(h-1) nodes m\*\*(h-1) \* (m-1) keys at level h

--------------------

m\*\*h - 1 keys total

**QUES:-**Which of the following sorting procedure is the slowest ?

A.Quick sort B.Heap sort C)Shell sort D)Bubble sort(yes)

***QUES:- The number of different binary trees with 6 nodes is \_\_\_\_\_\_.***

(1) 6 (2) 42 (3) 132 (4 ) 256

**Explanation:-** Formula is 

***QUES:-Match the following with respect to algorithm paradigms :***

a. Merge sort i. Dynamic programming(C)

b. Huffman coding ii. Greedy approach(B)

c. Optimal polygon triangulation iii. Divide and conquer(A)

d. Subset sum problem iv. Back tracking(D)

***QUES:-Red-black trees are one of many search tree schemes that are “balanced” in order to guarantee that basic dynamic-set operations take \_\_\_\_\_\_\_\_ time in the worst case.***

(1) O(1) (2) O(lg n)yes (3) O(n) (4) O(n lg n)

***QUES:-The time complexities of some standard graph algorithms are given. Match each algorithm with its time complexity ? (n and m are no. of nodes and edges respectively)***

a. Bellman Ford algorithm 1. O (m log n)

b. Kruskals algorithm 2. O (n3)

c. Floyd Warshall algorithm 3. O(mn)

d. Topological sorting 4. O(n + m)

**a b c d**

(A) 3 1 2 4 (yes)

(B) 2 4 3 1

(C) 3 4 1 2

(D) 2 1 3 4

***You have to sort a list L, consisting of a sorted list followed by a few ‘random’ elements. Which of the following sorting method would be most suitable for such a task ?***

(A) Bubble sort (B) Selection sort (C) Quick sort (D) Insertion sort(YES)

***Maximum number of edges in a n-Node undirected graph without self loop is***

(A) n2 (B) n(n – 1) (C) n(n + 1) (D)n(n – 1)/2(yes)

***A hash table has space for 75 records, then the probability of collision before the table is 6% full is?***

(A) .25 (B) .20 (C) .35 (D) .30

**Explanation:-**

To make the table 6% full, we need to insert at least ( 75 \* 0∙6 / 100) = 4.5 round up to 5 values.

Probability of collision during first insertion is 1/ 75

Probability of collision during third insertion is 2 /75

Probability of collision during fourth insertion is 3/ 75

Probability of collision during fifth insertion is 4 /75

Probability of collision during sixth insertion is 5 /75

So, total probability of collision to make the table 6% full is (1 + 2 + 3 + 4 + 5) /75 = 0.2

So option B is correct

***Which of the following data structure is Non-linear type ?***

(A) Strings (B) Lists (C) Stacks (D) None of the above(YES)

***The total number of comparisons in a bubble sort is***

(A) 0(log n) (B) 0(n log n) (C) 0(n) (D) None of the above(yes)

**Explanation:-**

First iteration of Outer loop ---------------> n-1 comparisions

Second iteration of Outer loop ----------> n-2 comparisions (Since largest element has gone to its proper place)

Third iteration of Outer loop--------------->n-3 comparisions (Since two largest elements has gone to their proper places)

Total number of comparisions = (n-1)+(n-2)+(n-3)+............+1 = n(n-1) / 2 = O(n^2)

***Which of the following is a bad example of recursion ?***

(A) Factorial (B) Fibonacci numbers(yes)

(C) Tower of Hanai (D) Tree traversal

***The run time complexity of Dijkstra’s algorithm to find the shortest path of a graph with n nodes is***

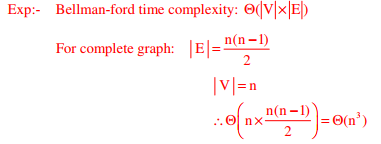
(A) O(n) (B) O(n - 1) (C) O(n^2) (yes) (D) O(nlogn)

Explanation:-

The time complexity of the above code/algorithm looks O(V^2) as there are two nested while loops. If we take a closer look, we can observe that O(ELogV)

***What is the time complexity of Bellman-Ford single-source shortest path algorithm on a complete graph of n vertices?***

A) Θ(n2) B) Θ(n2 log n) C) Θ(n3)(yes) D) Θ( m3 log n)



***The decision tree classifier is a widely used technique for \_\_\_\_\_\_.***

A.Association B.Classification(yes)

C.Clustering D.Partition

**Explanation:**

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

***The efficient data structure to insert/delete a number in a stored set of numbers is***

(A) Queue (B) Linked list

(C) Doubly linked list(yes) (D) Binary tree

***The amortized time complexity to perform \_\_\_\_\_\_ operation(s) in Splay trees is O(Ig n).***

(A) Search (B) Search and Insert

(C) Search and Delete (D) Search, Insert and Delete(YES)

***Consider the following statements : Which of the above statements is/are true ?***

(i) A graph in which there is a unique path between every pair of vertices is a tree.

(ii) A connected graph with e = v – 1 is a tree.

(iii) A graph with e = v – 1 that has no circuit is a tree.

(A) (i) & (iii) (B) (ii) & (iii) (C) (i) & (ii) (D) All of the above(yes)

***Linked Lists are not suitable for \_\_\_\_\_.***

(A) Binary Search(yes) (B) Polynomial Manipulation

(C) Insertion (D) Radix Sort

**Consider an undirected graph G with 100 nodes. The maximum number of edges to be included in G so that the graph is not connected is**

(A) 2451 (B) 4950 (C) 4851(yes) (D) 9900

Explanation:-

When maximum number of edges are added and still 100 node graph is disconnected means we made a 99 node complete graph and left one node disconnected.

No of edges in complete graph of x nodes = x(x-1) /2 = 99(99-1)/2 =4851

The n vertex graph with the maximal number of edges that is still disconnected is a Kn−1

a complete graph Kn−1 with n−1 vertices has (n−1)/2 edges, so ((n−1)(n−2))/2 + 1 edges.

***A simple graph G with n-vertices is connected if the graph has***

(A) (n – 1) (n – 2)/2 edges

(B) more than (n – 1) (n – 2)/2 edges(yes)

(C) less than (n – 1) (n – 2)/2 edges

(D) Σki=1 C(ni, 2) edges

**For a B-tree of height h and degree t, the total CPU time used to insert a node is**

(A) O(h log t) (B) O(t log h) (C) O(t^2h) (D) O(th)

**The time complexity to build a heap with a list of n numbers is (June – 2013 – PII)**

(A) O(log n) (B) O(n)(yes) (C) O(n logn) (D) O(n2)

***Consider the fractional knapsack instance n = 4, (p1, p2, p3, p4) = (10, 10, 12, 18), (w1, w2, w3, w4) = (2, 4, 6, 9) and M = 15. The maximum profit is given by (Assume p and w denotes profit and weight of objects respectively)***

(A) 40 (B) 38(yes) (C) 32 (D) 30

Explanation: w1+w2+w4 = 15 and its profit is 38

***Which one of the following is a physical data structure ?***

(A) Array(yes) (B) Linked lists (C) Stacks (D) Tables

Explanation:-

Because, Array is the only one which is going to implemented in memory exactly.

Stack is a logical model, which is implemented by array and linked list.

Table is a logical model, which is implemented by array.

Linked List is a logical model which is implemented by self-referential structure variable.

***Which algorithm has same average, worst case and best case time ?***

(A) Binary search (B) Maximum of n number(yes)

(C) Quick sort (D) Fibonacci search

***Let T(n) be the function defined by T(n)=1 and T(n)=2T(n/2)+√n, which of the following is TRUE ?***

(A) T(n)=O(√n) (B) T(n)=O(log2n) (C) T(n)=O(n)(yes) (D) T(n)=O(n2)

**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(yes)

**Explanation:-** Inorder traversal should alaways be in ascending order. So choose the result which is in ascending order

***A list of n strings, each of length n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is(Gate – 2012 )***

(A) O(n log n) (B) O(n2 log n) (C) O(n2 + log n) (D) O(n2)

**Explanation1:-**

When we are sorting an array of n integers, Recurrence relation for Total number of comparisons involved will be,

T(n) = 2T(n/2) + (n) where (n) is the number of comparisons in order to merge 2 sorted subarrays of size n/2.

= (nlog2n)

Instead of integers whose comparison take O(1) time, we are given n strings. We can compare 2 strings in O(n) worst case. Therefore, Total number of comparisons now will be (n2log2n) where each comparison takes O(n) time now.

In general, merge sort makes (nlog2n) comparisons, and runs in (nlog2n) time if each comparison can be done in O(1) time.

**Explanation2:-**

The recurrence tree for merge sort will have height Logn. And O(n2) work will be done at each level of the recurrence tree (Each level involves n comparisons and a comparison takes O(n) time in worst case). So time complexity of this Merge Sort will be O(n2 log n).

***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(NET – 2016 – PII)***

(1)n2–1 (2)n2+ 1 (3) (n – 1)/2 (yes) (4) (n + 1)/2

**The running time of an algorithm is O(g(n)) if and only if (JULY – 2020)**

(A) Its worst-case running time is O(g(n)) and its best-case running time is Ω(g(n)).

(O = bigO) (ANS)

(B) Its worst-case running time is Ω(g(n)) and its best-case running time is O(g(n)).

(O = big O)

(C) O(g(n)) = Ω (g(n))(O = big O)

(D) O(g(n)) ∩ ω(g(n)) is non-empty set, (o = small o)

Choose the correct answer from the options given below :

(1) (A) only(ANS) (2) (B) only (3) (C) only (4) (D) only

**In a binary max heap containing n numbers, the smallest element can be found in time (GATE CS 2006)**

(A) 0(n)(yes) (B) O(logn) (C) 0(loglogn) (D) 0(1)

Explanation:-

In a max heap, the smallest element is always present at a leaf node. So we need to check for all leaf nodes for the minimum value. Worst case complexity will be O(n)

***Which among the following statement(s) is (are) true? (June – 2020)***

***(A) A hash function takes a message of arbitrary length and generates a fixed length code.***

***(B) A hash function takes a message of fixed length and generates a code of variable length.***

***(C) A hash function may give same hash value for distinct messages***

Choose the correct answer from the options given below :

(1) (A) only (2) (B) and (C) only

(3) (A) and (C) only (yes) (4) (B) only

**Explanation:-**

Hash function is defined as any function that can be used to map data of arbitrary size of data to a fixed size data.. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes : Statement 1 is correct

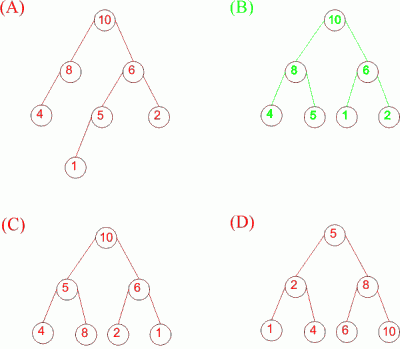
Yes, it is possible that a Hash Function maps a value to a same location in the memmory that’s why collision occurs and we have different technique to handle this problem : Statement 3 is coorect.

eg : we have hash function, h(x) = x mod 3

Acc to Statement 1, no matter what the value of ‘x’ is h(x) results in a fixed mapping location.

Acc. to Statement 3, h(x) can result in same mapping mapping location for different value of ‘x’ e.g. if x = 4 or x = 7 , h(x) = 1 in both the cases, although collision occurs.

***A max-heap is a heap where the value of each parent is greater than or equal to the value of its children. Which of the following is a max-heap? (Gate-2011)***



***Explanation:***

"n-element heap has height ⌊lg n⌋."

-Cormen (Introduction to Algorithms)

So, 7 elements heap has height 2. (A) has height 3 while (C) and (D) doesnot meet heap property.

***In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is (Gate - 2005)***

(A) nk (B) (n – 1)k + 1 (C) n(k – 1) + 1 (yes) (D) n(k – 1)

***A 5-ary tree is tree in which every internal node has exactly 5 children. The number of leaf nodes in such a tree with 8 internal nodes will be: (NET-July-2018)***

A. 30 B. 33 C. 45 D. 125

**Formula:**

L = I (n - 1) + 1

L =number of leaf nodes

I = number of internal nodes

n = n - ary tree

**Calculation:**

I = 8 n = 5

L = 8(5 - 1) + 1

L = 32 + 1 = 33

Number of leaf nodes = 33

Number of internal nodes = 8

Total number of nodes = 33 + 8 = 41

***In Activity-Selection problem, each activity i has a start time si and a finish time fi where si ≤ fi. Activities i and j are compatible if (Net-2015-Dec)***

1. Si ≥ fj
2. Sj ≥ fi
3. Si ≥ fj or sj ≥ fi (yes)
4. Si ≥ fj and  sj ≥ fi

Explanation:

Ref <https://www.studytonight.com/data-structures/activity-selection-problem>

***The following postfix expression is evaluated using a stack 823^/23\* + 51\* – The top two elements of the stack after first \* is evaluated (NET - 2012 - June)***

(A) 6, 1(Ans) (B) 5, 7 (C) 3, 2 (D) 1, 5

**Explanation:-**

Ref https://www.youtube.com/watch?v=u3paQa8KXu0&ab\_channel=Jenny%27slecturesCS%2FITNET%26JRF

***When using Dijkstra's algorithm to find shortest path in a graph, which of the following statements is not true?***

(A) It can find shortest path within the same graph data structure

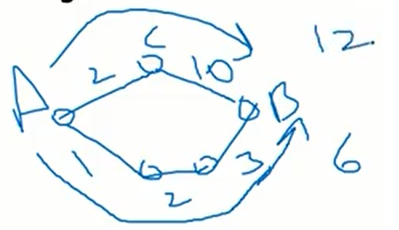
(B) Every time a new node is visited, we choose the node with smallest known distance/cost(weight) to visit first

(C) Shortest path always passes through least number of vertices(yes)

(D) The graph needs to have a non-negative weight on every edge

**Explanation**

**Ref** https://www.youtube.com/watch?v=\_v7g-rYM8hQ

****

Ref https://leetcode.com/problems/balanced-binary-tree/

**Two balanced binary trees are given with m and n elements, respectively. They can be merged into a balanced binary search tree in \_\_\_\_ time. (2021 – NOV)**

1. O(m\*log n)
2. O(m\*log(m+n))
3. O(m\*n)

4) O(m+n)

Answer Given:- O(m\*n)

***Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?***

f1(n) = 2^n

f2(n) = n^(3/2)

f3(n) = nLogn

f4(n) = n^(Logn)

(A) f3, f2, f4, f1

(B) f3, f2, f1, f4

(C) f2, f3, f1, f4

(D) f2, f3, f4, f1

**Answer: (A)**

**Explanation:** nLogn is the slowest growing function, then comes n^(3/2), then n^(Logn). Finally, 2^n is the fastest growing function.

Ref <https://testbook.com/question-answer/which-of-the-given-options-provides-the-increasing--60b619d707b467946838a25b>

***Two balanced binary trees are given with m and n elements, respectively. They can be merged into a balanced binary search tree in \_\_\_\_ time.(NET-2021)***

1. O(m\*log n), B) O(m\*log(m+n)), C) O(m\*n), D) O(m+n)

Answer Given:- O(m\*n)

**Of the following sorting algorithms, which has a running time that is least dependent on the initial ordering of the input? (ISRO - 2018)**

(A) Merge Sort(yes) (B) Insertion Sort (C) Selection Sort (D) Quick Sort

**Explanation:** In Insertion sort if the array is already sorted then it takes O(n) and if it is reverse sorted then it takes O(n2) to sort the array. In Quick sort, if the array is already sorted or if it is reverse sorted then it takes O(n2).The best and worst case performance of Selection is O(n2) only. But if the array is already sorted then less swaps take place. In merge sort, time complexity is O(nlogn) for all the cases and performance is affected least on the the order of input sequence.

***The preorder traversal is 12, 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20 then post order traversal of this tree is***

(A) 2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20

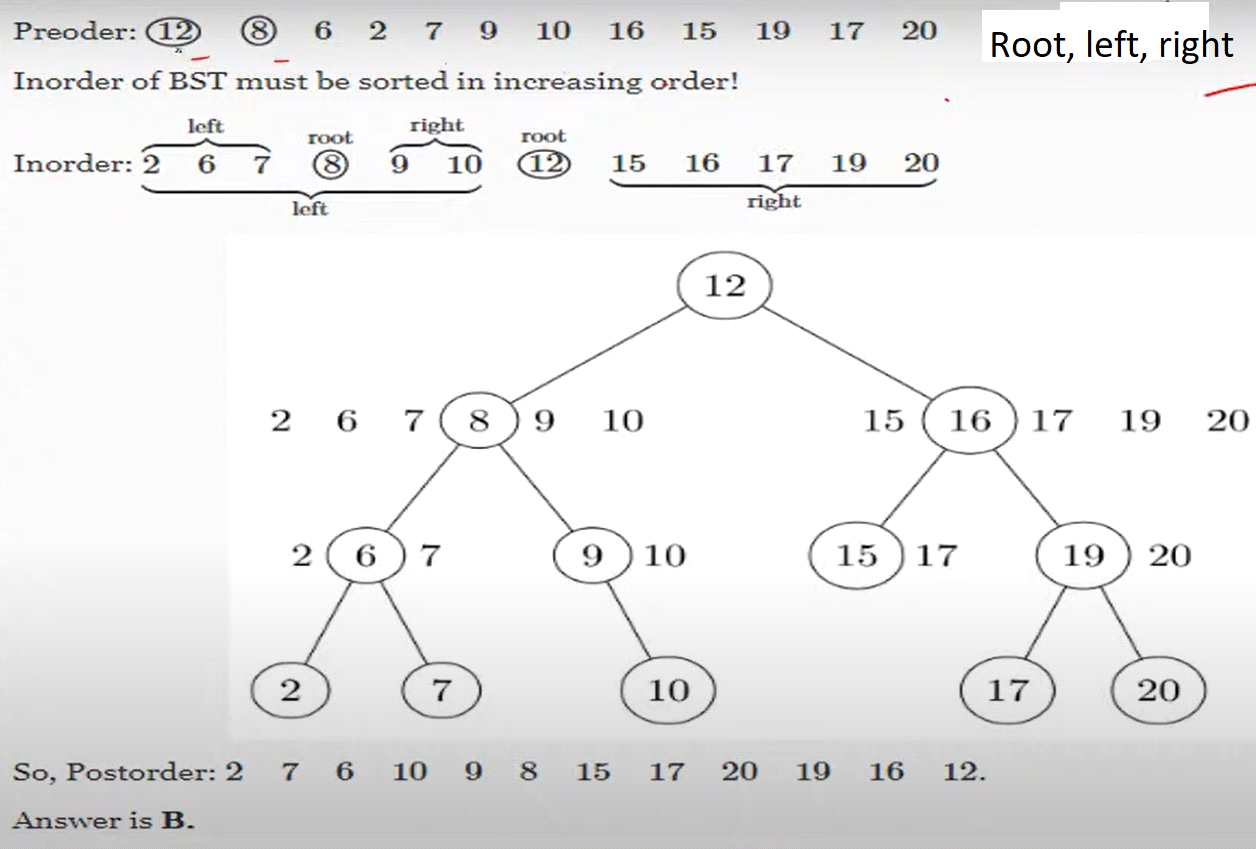
(B) 2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12

(C) 7, 2, 6, 8, 9, 10, 20, 17, 19, 15, 16, 12

(D) 7, 6, 2, 10, 9, 8, 15, 16, 17, 20, 19, 12

**Explanation:-**

Ref: <https://www.youtube.com/watch?v=-hEq5etPcNY&ab_channel=UnifyStudy-UnitedInformationforyou>



***Match List-I with List-II :***

**List-I List-II**

(a) Greedy best-first (i) Minimal cost (p) + h(p) (c)

(b) Lowest cost-first (ii) Minimal h(p) (b)

(c) A\* algorithm (iii) Minimal cost (p) (a)

Choose the correct option from those given below :

**Explanation:-**

h(p)=heuristic cost from current node to goal node

cost(p)=cost to reach that node

The worst case time complexity of Greedy best first search is O(bm). Where, m is the maximum depth of the search space.

The time and space complexity of A\* search algorithm is O(b^d) where b is the branching factor.

***Consider the following terminology and match List-I with List-Il and choose the correct answer from the code given below. (NET - 2018)***

b = branching factor

d = depth of the shallowest solution

m = Maximum depth of the search tree

I = depth limit

**List-I (Algorithms) List-II (Space Complexity)**

(a) BFS search (i) O(bd)

(b) DFS search (ii) O(bd)

(c) Depth-limited search (iii) O(bm)

(d) Iterative deepening search (iv) O(bl)