Question Bank

CSD 102

1. The minimum number of comparisons required to determine if an integer appears more than n/2 times in a sorted array of n integers is
   A. Θ(n)
   B. Θ(logn)
   C. Θ(n*logn)
   D. Θ(1)

2. Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

```
C = 100
for i = 1 to n do
  for j = 1 to n do
  {
    Temp = A[i][j] + C
    A[i][j] = A[j][i]
    A[j][i] = Temp - C
  }
for i = 1 to n do
  for j = 1 to n do
    Output(A[i][j]);
```

   A. The matrix A itself
   B. Transpose of matrix A
   C. Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A
   D. None of the above

3. Consider a two dimensional array A[20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ?
   A. 560
   B. 460
   C. 570
   D. 575

4. An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by k places, where 1 <= k <= (n-1). An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume alt the variables are suitably declared.

```
min = n; i = 0;
while (_____) {
   temp = A[i]; j = i;
   while (_____) {
   A[j] = _____
   j= (j + k) mod n ;
   If ( j< min ) then
      min = j;
}
A[(n + i — k) mod n] = _____
i = _____
```

A. i > min; j!= (n+i)mod n; A[j + k]; temp; i + 1 ;
B. i < min; j!= (n+i)mod n; A[j + k]; temp; i + 1;
C. i > min; j!= (n+i+k)mod n; A[(j + k)]; temp; i + 1;
D. i < min; j!= (n+i-k)mod n; A[(j + k)mod n]; temp; i + 1

5. Consider the following function that takes reference to head of a Doubly Linked List as parameter. Assume that a node of doubly linked list has previous pointer as prev and next pointer as next.

```
void fun(struct node **head_ref)
{
   struct node *temp = NULL;
   struct node *current = *head_ref;
   while (current !=  NULL)
   {
      temp = current->prev;
      current->prev = current->next;
      current->next = temp;
      current = current->prev;
   }
   if(temp != NULL )
      *head_ref = temp->prev;
}
```

Assume that reference of head of following doubly linked list is passed to above function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call?

A. 2 <--> 1 <--> 4 <--> 3 <--> 6 <-->5
B. 5 <--> 4 <--> 3 <--> 2 <--> 1 <-->6
C. 6 <--> 5 <--> 4 <--> 3 <--> 2 <--> 1
D. 6 <--> 5 <--> 4 <--> 3 <--> 1 <--> 2

6. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function.

```
/* Link list node */
struct node
{
   int data;
   struct node* next;
};

/* head_ref is a double pointer which points to head (or start) pointer
 of linked list */
static void reverse(struct node** head_ref)
{
   struct node* prev   = NULL;
   struct node* current = *head_ref;
   struct node* next;
   while (current != NULL)
   {
      next  = current->next;
      current->next = prev;
      prev = current;
      current = next;
   }
   /*ADD A STATEMENT HERE*/
}
```

What should be added in place of "/*ADD A STATEMENT HERE*/", so that the function correctly reverses a linked list.

A. *head_ref = prev;
B. *head_ref = current;
C. *head_ref = next;
D. *head_ref = NULL;

7. What is the output of following function in which start is pointing to the first node of the following linked list 1->2->3->4->5->6 ?

```
void fun(struct node* start)
{
 if(start == NULL)
  return;
 printf("%d  ", start->data);

 if(start->next != NULL )
  fun(start->next->next);
 printf("%d  ", start->data);
```

}

A. 1 4 6 6 4 1
B. 1 3 5 1 3 5
C. 1 2 3 5
D. 1 3 5 5 3 1

8. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is
   A. log(2*n)
   B. n/2
   C. log(2*n) -1
   D. n

9. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?
   A. rear node
   B. front node
   C. not possible with a single pointer
   D. node next to front

10. What are the time complexities of finding 8th element from beginning and 8th element from end in a singly linked list? Let n be the number of nodes in linked list, you may assume that n > 8.
    A. O(1) and O(n)
    B. O(1) and O(1)
    C. O(n) and O(1)
    D. O(n) and O(n)

11. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?

    A. Possible if X is not last node. Use following two steps (a) Copy the data of next of X to X. (b)Update the pointer of node X to the node after the next node. Delete next of X.
    B. Possible if size of linked list is even.
    C. Possible if size of linked list is odd
    D. Possible if X is not first node. Use following two steps (a) Copy the data of next of X to X. (b) Delete next of X.

12. The concatenation of two lists is to be performed in O(1) time. Which of the following implementations of a list should be used?
    A. singly linked list
    B. doubly linked list

C. circular doubly linked list
D. array implementation of lists

13. Consider the following statements:
    i. First-in-first out types of computations are efficiently supported by STACKS.
    ii. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
    iii. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
    iv. Last-in-first-out type of computations are efficiently supported by QUEUES.
    Which of the following is correct?

    A. (ii) is true
    B. (i) and (ii) are true
    C. (iii) is true
    D. (ii) and (iv) are true

14. Suppose there are two singly linked lists both of which intersect at some point and become a single linked list. The head or start pointers of both the lists are known, but the intersecting node and lengths of lists are not known. What is worst case time complexity of optimal algorithm to find intersecting node from two intersecting linked lists?
    A. Θ(n*m), where m, n are lengths of given lists
    B. Θ(n^2), where m>n and m, n are lengths of given lists
    C. Θ(m+n), where m, n are lengths of given lists
    D. Θ(min(n, m)), where m, n are lengths of given lists

15. Consider an implementation of unsorted single linked list. Suppose it has its representation with a head and a tail pointer (i.e. pointers to the first and last nodes of the linked list). Given the representation, which of the following operation can not be implemented in O(1) time ?
    A. Insertion at the front of the linked list.
    B. Insertion at the end of the linked list.
    C. Deletion of the front node of the linked list.
    D. Deletion of the last node of the linked list.

16. Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

```
void fun(int n)
{
  Stack S;  // Say it creates an empty stack S
  while (n > 0)
  {
    // This line pushes the value of n%2 to stack S
    push(&S, n%2);
    n = n/2;
```

```
    }
    // Run while Stack S is not empty
    while (!isEmpty(&S))
     printf("%d ", pop(&S)); // pop an element from S and print it
  }
```

What does the above function do in general?

A. Prints binary representation of n in reverse order
B. Prints binary representation of n
C. Prints the value of Logn
D. Prints the value of Logn in reverse order

17. Which one of the following is an application of Stack Data Structure?
    A. Managing function calls
    B. The stock span problem
    C. Arithmetic expression evaluation
    D. All of the above

18. Which of the following is true about linked list implementation of stack?
    A. In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
    B. In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
    C. Both of the above
    D. None of the above

19. Following is an incorrect pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while (more input is available)
{
   read a character
   if (the character is a '(' )
      push it on the stack
   else if (the character is a ')' and the stack is not empty )
      pop a character off the stack
   else
      print "unbalanced" and exit
 }
 print “balanced”
```

Which of these unbalanced sequences does the above code think is balanced?

A. ((())
B. ())(()
C. (()()))
D. ((()))()

20. The following postfix expression with single digit operands is evaluated using a stack:

    8 2 3 ^ / 2 3 * + 5 1 * -

    Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

    A. 6, 1
    B. 5, 7
    C. 3, 2
    D. 1, 5

21. Assume that the operators +, -, × are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, * , +, -. The postfix expression corresponding to the infix expression a + b × c - d ^ e ^ f is

    A. abc × + def ^ ^ -
    B. abc × + de ^ f ^ -
    C. ab + c × d - e ^ f ^
    D. - + a × bc ^ ^ def

22. The result evaluating the postfix expression 10 5 + 60 6 / * 8 – is

    A. 284
    B. 213
    C. 142
    D. 71

23. If the sequence of operations - push (1), push (2), pop, push (1), push (2), pop, pop, pop, push (2), pop are performed on a stack, the sequence of popped out values

    A. 2,2,1,1,2
    B. 2,2,1,2,2
    C. 2,1,2,2,1
    D. 2,1,2,2,2

24. Stack A has the entries a, b, c (with a on top). Stack B is empty. An entry popped out of stack A can be printed immediately or pushed to stack B. An entry popped out of the stack B can be only be printed. In this arrangement, which of the following permutations of a, b, c are not possible?

    A. b a c
    B. b c a
    C. c a b
    D. a b c

25. Convert the following infix expression into its equivalent post fix expression (A + B^ D) / (E – F) + G

    A. ABD^ + EF – / G+
    B. ABD + ^EF – / G+

C. ABD + ^EF / – G+
D. ABD^ + EF / – G+

26. Following is C like pseudo-code of a function that takes a Queue as an argument, and uses a stack S to do processing.

```
void fun(Queue *Q)
{
    Stack S;  // Say it creates an empty stack S
    // Run while Q is not empty
    while (!isEmpty(Q))
    {
        // deQueue an item from Q and push the dequeued item to S
        push(&S, deQueue(Q));
    }
    // Run while Stack S is not empty
    while (!isEmpty(&S))
    {
      // Pop an item from S and enqueue the popped item to Q
      enQueue(Q, pop(&S));
    }
}
```

What does the above function do in general?

A. Removes the last from Q
B. Keeps the Q same as it was before the call
C. Makes Q empty
D. Reverses the Q

27. A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements ”1‘ and ”7‘ are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

A. 10, 8, 7, 5, 3, 2, 1
B. 10, 8, 7, 2, 3, 1, 5
C. 10, 8, 7, 1, 2, 3, 5
D. 10, 8, 7, 3, 2, 1, 5

28. Consider the following pseudo code. Assume that IntQueue is an integer queue. What does the function fun do?

```
void fun(int n)
{
    IntQueue q = new IntQueue();
    q.enqueue(0);
    q.enqueue(1);
    for (int i = 0; i < n; i++)
    {
```

```
        int a = q.dequeue();
        int b = q.dequeue();
        q.enqueue(b);
        q.enqueue(a + b);
        print(a);
    }
}
```

A. Prints numbers from 0 to n-1
B. Prints numbers from n-1 to 0
C. Prints first n Fibonacci numbers
D. Prints first n Fibonacci numbers in reverse order

29. Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:
i. isEmpty (Q) — returns true if the queue is empty, false otherwise.
ii. delete (Q) — deletes the element at the front of the queue and returns its value.
iii. insert (Q, i) — inserts the integer i at the rear of the queue.
Consider the following function:

```
void f (queue Q) {
int i ;
if (!isEmpty(Q)) {
   i = delete(Q);
   f(Q);
   insert(Q, i);
  }
}
```

What operation is performed by the above function f ?

A. Leaves the queue Q unchanged
B. Reverses the order of the elements in the queue Q
C. Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order
D. Empties the queue Q

30. Consider a standard Circular Queue 'q' implementation (which has the same condition for Queue Full and Queue Empty) whose size is 11 and the elements of the queue are q[0], q[1], q[2].....,q[10]. The front and rear pointers are initialized to point at q[2] . In which position will the ninth element be added?
A. q[0]
B. q[1]
C. q[9]
D. q[10]

31. Level of a node is distance from root to that node. For example, level of root is 1 and levels of left and right children of root is 2. The maximum number of nodes on level i of a binary tree is

In the following answers, the operator '^' indicates power.

A. 2^(i-1)
B. 2^i
C. 2^(i+1)
D. 2^[(i+1)/2]

32. In a complete k-ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is:
    A. nk
    B. (n – 1) k+ 1
    C. n(k – 1) + 1
    D. n(k – 1)

33. The maximum number of binary trees that can be formed with three unlabelled nodes is:
    A. 1
    B. 5
    C. 4
    D. 3

34. Postorder traversal of a given binary search tree, T produces the following sequence of keys 10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29 Which one of the following sequences of keys can be the result of an in-order traversal of the tree T?
    A. 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95
    B. 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29
    C. 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95
    D. 95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29

35. Consider a node X in a Binary Tree. Given that X has two children, let Y be Inorder successor of X. Which of the following is true about Y?
    A. Y has no right child
    B. Y has no left child
    C. Y has both children
    D. None of the above\

36. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:
    A. 2^h-1
    B. 2^{h-1} -1
    C. 2^{h+1}-1
    D. 2^{h+1}

37. Consider a complete binary tree where the left and the right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is
    A. Ω(logn)

B. Ω(n)
C. Ω(nlogn)
D. Ω(n2)

38. An array of integers of size n can be converted into a heap by adjusting the heaps rooted at each internal node of the complete binary tree starting at the node ⌊(n - 1) /2⌋, and doing this adjustment up to the root node (root node is at index 0) in the order ⌊(n - 1)/2⌋, ⌊(n - 3)/ 2⌋, ..., 0. The time required to construct a heap in this manner is
    A. O(log n)
    B. O(n)
    C. O (n log log n)
    D. O(n log n)

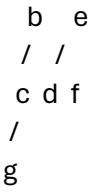
39. Breadth First Search (BFS) is started on a binary tree beginning from the root vertex. There is a vertex t at a distance four from the root. If t is the n-th vertex in this BFS traversal, then the maximum possible value of n is ___31_____.

40. In a binary tree, the number of internal nodes of degree 1 is 5, and the number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is
    A. 10
    B. 11
    C. 12
    D. 15

41. The following three are known to be the preorder, inorder and postorder sequences of a binary tree. But it is not known which is which.
    MBCAFHPYK
    KAMCBYPFH
    MABCKYFPH
    Pick the true statement from the following.
    A. I and II are preorder and inorder sequences, respectively
    B. I and III are preorder and postorder sequences, respectively
    C. II is the inorder sequence, but nothing more can be said about the other two sequences
    D. II and III are the preorder and inorder sequences, respectively

42. Draw the binary tree with node labels a, b, c, d, e, f and g for which the inorder and postorder traversals result in the following sequences:
    Inorder a f b c d g e
    Postorder a f c g e d b

43. Which of the following sequences denotes the post order traversal sequence of the given tree?



A. f e g c d b a
B. g c b d a f e
C. g c d b f e a
D. f e d g c b a

44. 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:
    A. ceiling (i/2)
    B. floor (i/2)
    C. ceiling ((i+1)/2)
    D. floor ((i+1)/2)

45. A strictly binary tree with 10 leaves
    A. cannot have more than 19 nodes
    B. has exactly 19 nodes
    C. has exactly 17 nodes
    D. has exactly 20 nodes

46. The number of structurally different possible binary trees with 4 nodes is
    A. 14
    B. 12
    C. 336
    D. 168

47. What is the maximum height of any AVL tree with 7 nodes? Assume that height of tree with single node is 0.
    A. 2
    B. 3
    C. 4
    D. 5

48. The in-order and pre-order traversal of a binary tree are d b e a f c g and a b d e c f g respectively. The post order traversal of a binary tree is
    A. e d b g f c a
    B. e d b f g c a
    C. d e b f g c a
    D. d e f g b c a

49. A complete binary tree with n non-leaf nodes contains
    A. log2 n nodes
    B. n+1 nodes
    C. 2n nodes
    D. 2n+1 nodes

50. 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

51. A complete binary tree with the property that the value at each node is as least as large as the values at its children is known as
    A. Binary search tree
    B. AVL tree
    C. Completely balanced tree
    D. Heap

52. What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree for a skewed tree ?
    A. O(n) for all
    B. O(Logn) for all
    C. O(Logn) for search and insert, and O(n) for delete
    D. O(Logn) for search, and O(n) for insert and delete

53. In delete operation of BST, we need inorder successor (or predecessor) of a node when the node to be deleted has both left and right child as non-empty. Which of the following is true about inorder successor needed in delete operation?
    A. Inorder Successor is always a leaf node
    B. Inorder successor is always either a leaf node or a node with empty left child
    C. Inorder successor may be an ancestor of the node
    D. Inorder successor is always either a leaf node or a node with empty right child

54. We are given a set of n distinct elements and an unlabelled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?
    A. 0
    B. 1
    C. n!
    D. (1/(n+1)).2nCn

55. How many distinct binary search trees can be created out of 4 distinct keys?

A. 4
B. 14
C. 24
D. 42

56. Which of the following traversal outputs the data in sorted order in a BST?
    A. Preorder
    B. Inorder
    C. Postorder
    D. Level order

57. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?
    A. 7 5 1 0 3 2 4 6 8 9
    B. 0 2 4 3 1 6 5 9 8 7
    C. 0 1 2 3 4 5 6 7 8 9
    D. 9 8 6 4 2 3 0 1 5 7

58. 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 (the height is the maximum distance of a leaf node from the root)?
    A. 2
    B. 3
    C. 4
    D. 6

59. The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the same tree?
    A. 10, 20, 15, 23, 25, 35, 42, 39, 30
    B. 15, 10, 25, 23, 20, 42, 35, 39, 30
    C. 15, 20, 10, 23, 25, 42, 35, 39, 30
    D. 15, 10, 23, 25, 20, 35, 42, 39, 30

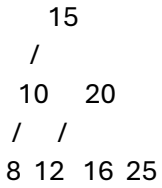
60. Which of the following traversals is sufficient to construct BST from given traversals 1) Inorder 2) Preorder 3) Postorder
    A. Any one of the given three traversals is sufficient
    B. Either 2 or 3 is sufficient
    C. 2 and 3
    D. 1 and 3

61. Consider the following code snippet in C. The function print() receives root of a Binary Search Tree (BST) and a positive integer k as arguments.

```
// A BST node
```

```
struct node {
   int data;
   struct node *left, *right;
};

int count = 0;

void print(struct node *root, int k)
{
   if (root != NULL && count <= k)
   {
      print(root->right, k);
      count++;
      if (count == k)
        printf("%d ", root->data);
      print(root->left, k);
   }
}
```

What is the output of print(root, 3) where root represent root of the following BST.



A. 10
B. 16
C. 20
D. 20 10

62. Consider the same code as given in above question. What does the function print() do in general? The function print() receives root of a Binary Search Tree (BST) and a positive integer k as arguments.

```
// A BST node
struct node {
   int data;
   struct node *left, *right;
};

int count = 0;

void print(struct node *root, int k)
{
```

```
    if (root != NULL && count <= k)
    {
       print(root->right, k);
       count++;
       if (count == k)
         printf("%d ", root->data);
       print(root->left, k);
    }
}
```

A. Prints the kth smallest element in BST
B. Prints the kth largest element in BST
C. Prints the leftmost node at level k from root
D. Prints the rightmost node at level k from root

63. You are given the postorder traversal, P, of a binary search tree on the n elements 1, 2, ..., n. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?
    A. O(Logn)
    B. O(n)
    C. O(nLogn)
    D. none of the above, as the tree cannot be uniquely determined.

64. While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is
    A. 65
    B. 67
    C. 69
    D. 83

65. The number of ways in which the numbers 1, 2, 3, 4, 5, 6, 7 can be inserted in an empty binary search tree, such that the resulting tree has height 6, is _____64_____ Note: The height of a tree with a single node is 0.

66. When searching for the key value 60 in a binary search tree, nodes containing the key values 10, 20, 40, 50, 70 80, 90 are traversed, not necessarily in the order given. How many different orders are possible in which these key values can occur on the search path from the root to the node containing the value 60?
    A. 35
    B. 64
    C. 128
    D. 5040

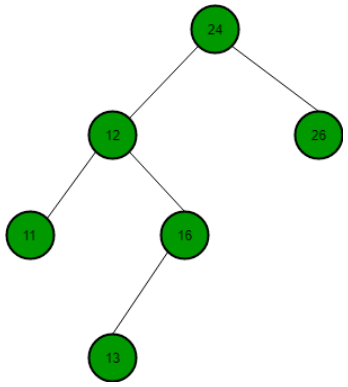
67. A binary search tree is generated by inserting in order the following integers:
    50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

The number of nodes in the left subtree and right subtree of the root respectively is

A. (4, 7)
B. (7, 4)
C. (8, 3)
D. (3, 8)

68. The postorder traversal of a binary tree is 8, 9, 6, 7, 4, 5, 2, 3, 1. The inorder traversal of the same tree is 8, 6, 9, 4, 7, 2, 5, 1, 3. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is ____4____.

69. 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

70. 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. $\theta(\log_t n)$

71. Following function is supposed to calculate the maximum depth or height of a Binary tree -- the number of nodes along the longest path from the root node down to the farthest leaf node.

```
int maxDepth(struct node* node)
{
   if (node==NULL)
       return 0;
   else
   {
       /* compute the depth of each subtree */
```

```
        int lDepth = maxDepth(node->left);
        int rDepth = maxDepth(node->right);

        /* use the larger one */
        if (lDepth > rDepth)
            return X;
        else return Y;
    }
}
```

What should be the values of X and Y so that the function works correctly?

A. X = lDepth, Y = rDepth
B. X = lDepth + 1, Y = rDepth + 1
C. X = lDepth - 1, Y = rDepth -1
D. None of the above

72. What is common in three different types of traversals (Inorder, Preorder and Postorder)?
    A. Root is visited before right subtree
    B. Left subtree is always visited before right subtree
    C. Root is visited after left subtree
    D. All of the above
    E. None of the above

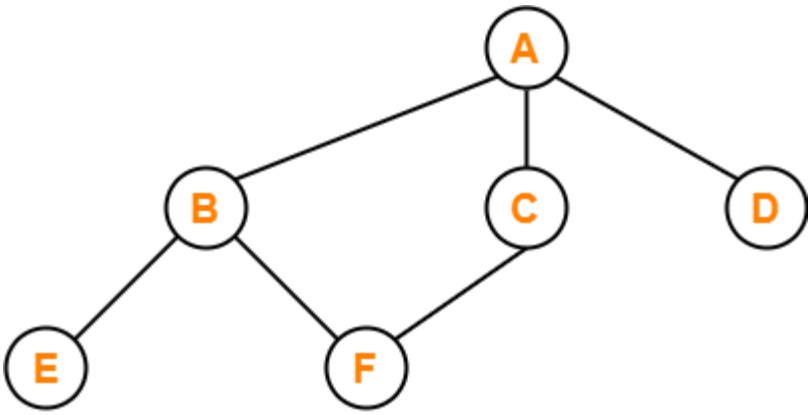
73. What does the following function do for a given binary tree?

```
int fun(struct node *root)
{
   if (root == NULL)
      return 0;
   if (root->left == NULL && root->right == NULL)
      return 0;
   return 1 + fun(root->left) + fun(root->right);
}
```

    A. Counts leaf nodes
    B. Counts internal nodes
    C. Returns height where height is defined as number of edges on the path from root to deepest node
    D. Return diameter where diameter is number of edges on the longest path between any two nodes.

74. Which of the following pairs of traversals is not sufficient to build a binary tree from the given traversals?
    A. Preorder and Inorder
    B. Preorder and Postorder
    C. Inorder and Postorder
    D. None of the Above

75. Consider the following graph-



Breadth First Search Example

What is the breadth first search traversal order of the above graph for starting vertex as A?

76. Which traversal of tree resembles the breadth first search of the graph?
    A. Preorder
    B. Inorder
    C. Postorder
    D. Level order

77. Consider the following C program segment

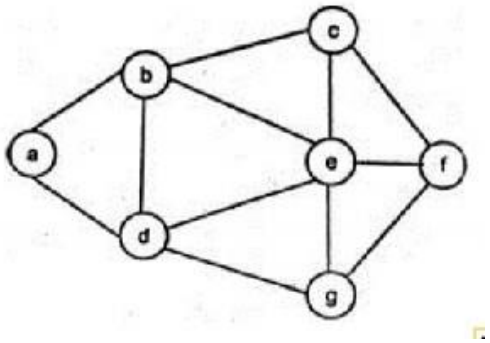
```
struct CellNode
{
  struct CelINode *leftchild;
  int element;
  struct CelINode *rightChild;
}

int Dosomething(struct CelINode *ptr)
{
    int value = 0;
    if (ptr != NULL)
    {
      if (ptr->leftChild != NULL)
        value = 1 + DoSomething(ptr->leftChild);
      if (ptr->rightChild != NULL)
        value = max(value, 1 + DoSomething(ptr->rightChild));
    }
    return (value);
}
```

The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is

A. The number of leaf nodes in the tree
B. The number of nodes in the tree
C. The number of internal nodes in the tree
D. The height of the tree

78. The array representation of a complete binary tree contains the data in sorted order. Which traversal of the tree will produce the data in sorted form?
A. Preorder
B. Inorder
C. Postorder
D. Level order

79. Consider the following sequence of nodes for the undirected graph given below.
a b e f d g c
a b e f c g d
a d g e b c f
a d b c g e f
A Depth First Search (DFS) is started at node a. The nodes are listed in the order they are first visited. Which all of the above is (are) possible output(s)?



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

80. The number of rotations required to insert a sequence of elements 9,6,5,8,7,10 into an empty AVL tree is?
A. 0
B. 1
C. 2
D. 3

81. Assume that the operators +, −, × are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, ×, +, −. The postfix expression corresponding to the infix expression is

a + b × c − d ^ e ^ f

A. abc x + def ^ ^ –
B. abc x + de ^ f ^ –
C. ab + c × d – e^f^
D. – + a × b c^^ def

82. Consider a B+-tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node?
A. 1
B. 2
C. 3
D. 4

83. Which one of the following is a key factor for preferring B-trees to binary search trees for indexing database relations?
A. Database relations have a large number of records
B. Database relations are sorted on the primary key
C. B-trees require less memory than binary search trees
D. Data transfer form disks is in blocks.

84. 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

85. B+ Trees are considered BALANCED because
A. the lengths of the paths from the root to all leaf nodes are all equal.
B. the lengths of the paths from the root to all leaf nodes differ from each other by at most 1.
C. the number of children of any two non-leaf sibling nodes differ by at most 1.
D. the number of records in any two leaf nodes differ by at most 1

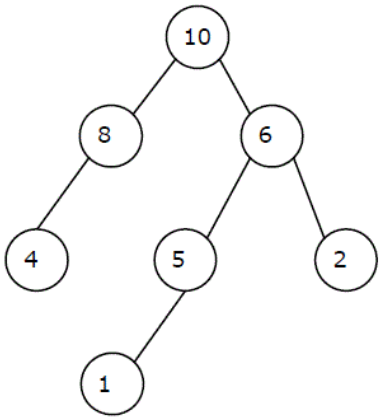
86. What is the time complexity of Build Heap operation? (Build Heap is used to build a max(or min) binary heap from a given array. Build Heap is used in Heap Sort as a first step for sorting.)
A. O(nLogn)
B. O(n^2)
C. O(n)
D. O(logn)

87. Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?

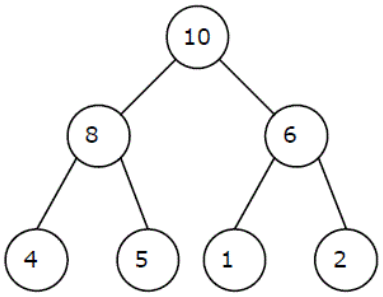
A. 1
B. 2
C. 3 or 4
D. 5 or 6

88. A max-heap is a heap where the value of each parent is greater than or equal to the values of its children. Which of the following is a max-heap?

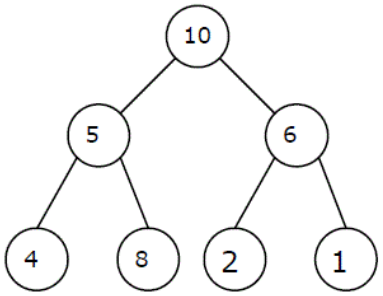
(A)



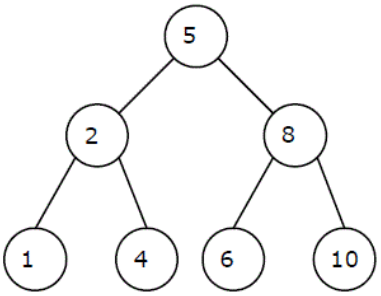
(B)



(C)



(D)



A. A
B. B
C. C
D. D

89. A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, a[0], nodes in the next level, from left to right, is stored from a[1] to a[3]. The nodes from the second level of the tree from left to right are stored from a[4] location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a[n] and pushing it up the tree to satisfy the heap property. Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?
A. 1, 3, 5, 6, 8, 9
B. 9, 6, 3, 1, 8, 5
C. 9, 3, 6, 8, 5, 1
D. 9, 5, 6, 8, 3, 1

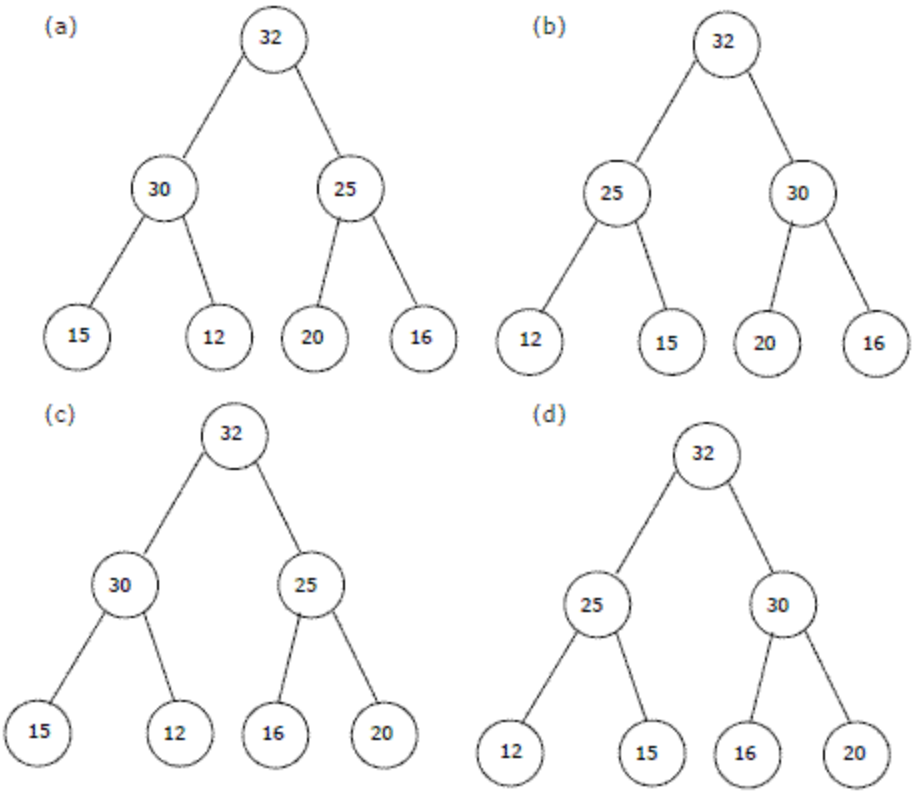
90. Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3- ary max heap found in the above question, Which one of the following is the sequence of items in the array representing the resultant heap?
    A. 10, 7, 9, 8, 3, 1, 5, 2, 6, 4
    B. 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
    C. 10, 9, 4, 5, 7, 6, 8, 2, 1, 3
    D. 10, 8, 6, 9, 7, 2, 3, 4, 1, 5

91. Consider a binary max-heap implemented using an array. Which among the following arrays represents a binary max-heap? (More than one option correct)
    A. 25,12,16,8,10,13,14
    B. 25,12,16,13,10,8,14
    C. 25,14,16,13,10,8,12
    D. 25,14,12,13,10,8,16

92. In a min-heap with n elements with the smallest element at the root, the 7th smallest element can be found in time:
    A. theta(nlogn)
    B. theta(n)
    C. theta(logn)
    D. theta(1)

93. What are the necessary condition for a Tree to be a heap?
    A. the tree must be complete.
    B. Every Root value is greater or smaller than the children's value.
    C. Both A and B
    D. None

94. The elements 32, 15, 20, 30, 12, 25, 16 are inserted one by one in the given order into a Max Heap. The resultant Max Heap is.



A. a
B. b
C. c
D. d

95. A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:
A. 10, 8, 7, 3, 2, 1, 5
B. 10, 8, 7, 2, 3, 1, 5
C. 10, 8, 7, 1, 2, 3, 5
D. 10, 8, 7, 5, 3, 2, 1

96. Consider a max heap, represented by the array: 40, 30, 20, 10, 15, 16, 17, 8, 4. Now consider that a value 35 is inserted into this heap. After insertion, the new heap is
A. 40, 30, 20, 10, 15, 16, 17, 8, 4, 35
B. 40, 35, 20, 10, 30, 16, 17, 8, 4, 15
C. 40, 30, 20, 10, 35, 16, 17, 8, 4, 15
D. 40, 35, 20, 10, 15, 16, 17, 8, 4, 30

97. Consider the following array of elements. ⟨89, 19, 50, 17, 12, 15, 2, 5, 7, 11, 6, 9, 100⟩. The minimum number of interchanges needed to convert it into a max-heap is

A. 4
B. 5
C. 2
D. 3

98. The minimum number of interchanges needed to convert the array 89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70 into a heap with the maximum element at the root is
A. 0
B. 1
C. 2
D. 3

99. How much time a heap will take to Find Min/Max value?
A. O(logN)
B. O(N)
C. O(1)
D. None

100. What is the maximum number of edges in an acyclic undirected graph with n vertices?
A. n-1
B. n
C. n + 1
D. 2n-1

101. Which of the following data structure is useful in traversing a given graph by breadth first search?
A. Stack
B. List
C. Queue
D. None of the above.

102. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

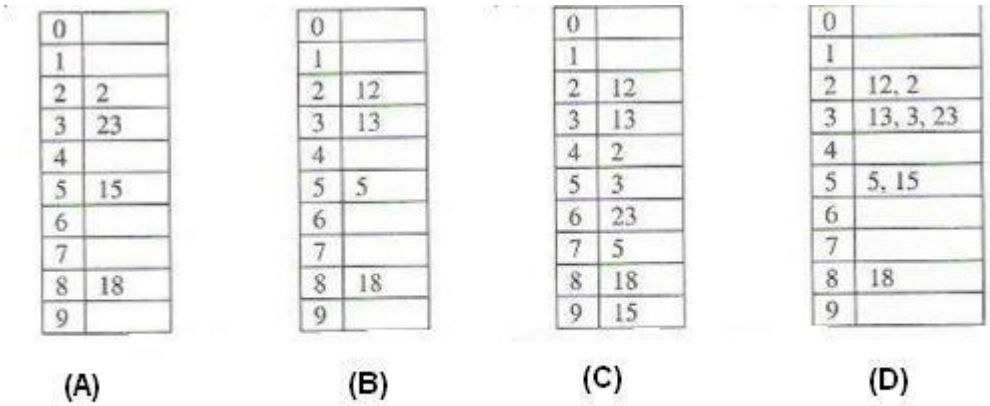
A. 46, 42, 34, 52, 23, 33
B. 34, 42, 23, 52, 33, 46
C. 46, 34, 42, 23, 52, 33
D. 42, 46, 33, 23, 34, 52

103. How many different insertion sequences of the key values using the hash function h(k) = k mod 10 and linear probing will result in the hash table shown below?

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

A. 10
B. 20
C. 30
D. 40

104. The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?



0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(A)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(C)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

(D)

A. A
B. B
C. C
D. D

105. Which searching technique takes O(1) time complexity for searching the data?
A. Binary Search
B. Linear Search
C. AVL Tree Search
D. Hashing

106. Consider a 13 element hash table for which f(key)=key mod 13 is used with integer keys. Assuming linear probing is used for collision resolution, at which location would the key 103 be inserted, if the keys 661, 182, 24 and 103 are inserted in that order?
A. 0
B. 1
C. 11
D. 12

107. Assume that the algorithms considered here sort the input sequences in ascending order. If the input is already in ascending order, which of the following are TRUE?
I. Quicksort runs in Θ(n2) time
II. Bubble sort runs in Θ(n2) time
III. Mergesort runs in Θ(n) time
IV. Insertion sort runs in Θ(n) time

A. I and II only
B. I and III only
C. II and IV only
D. I and IV only

108. Which one of the following in place sorting algorithms needs the minimum number of swaps?

A. Quick Sort
B. Insertion Sort
C. Selection Sort
D. Heap Sort

109. What is the number of swaps required to sort n elements using selection sort, in the worst case?

A. O(n)
B. O(nlogn)
C. O(n2)
D. O(n2logn)

110. Which of the following sorting algorithms has the lowest worst-case complexity?

A. Merge Sort
B. Bubble Sort
C. Quick Sort
D. Selection Sort

111. Which one of the following is the tightest upper bound that represents the time complexity of inserting an object into a binary search tree of n nodes?

A. O(1)
B. O(log n)
C. O(n)
D. O(n log n)

112. Consider the program below in a hypothetical language which allows global variable and a choice of call by reference or call by value methods of parameter passing.

```
int i ;
program main ()
{
  int j = 60;
  i = 50;
  call f (i, j);
  print i, j;
}
procedure f (x, y)
{
  i = 100;
  x = 10;
  y = y + i ;
}
```

Which one of the following options represents the correct output of the program for the two parameter passing mechanisms?

A. Call by value : i = 70, j = 10; Call by reference : i = 60, j = 70
B. Call by value : i = 50, j = 60; Call by reference : i = 50, j = 70
C. Call by value : i = 10, j = 70; Call by reference : i = 100, j = 60
D. Call by value : i = 100, j = 60; Call by reference : i = 10, j = 70

113. What does the given program print?

```
char c[ ] = "GATE2011"
char *p = c;
printf ("%s", p + p[3] – p[1]);
```

A. GATE 2011
B. 2011
C. E2011
D. 011

114. The output of the following C program is_____

```
void f1(int a, int b) {
        int c;
        c=a; a=b; b=c;
}
void f2(int *a, int *b) {
        int c;
        c=*a; *a=*b; *b=c;
}
int main(){
        int a=4, b=5, c=6;
        f1(a,b);
        f2(&b, &c);
        printf("%d",c-a-b);
}
```

A. -5
B. 6
C. -6
D. 0

115. The following program prints _____

```
#include < stdio.h >
void f (int *p, int *q) {
        p = q;
        *p = 2;
```

```
}
int i = 0, j = 1;
int main ( ){
        f(&i, &j);
        printf ("%d %d \ n", i, j);
        return 0;
}
```

A. 2 2
B. 2 1
C. 0 1
D. 0 2

116. The output of the following C program is

```
#include < stdio.h >
int main () {
        int arr [] = {1,2,3,4,5,6,7,8,9,0,1,2,5}, *ip = arr+4;
        printf ("%d\n", ip[1]);
        return 0;
}
```

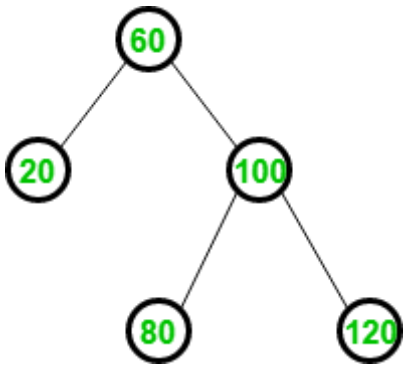
A. 6
B. 5
C. 66
D. 0

117. Consider the following C program:

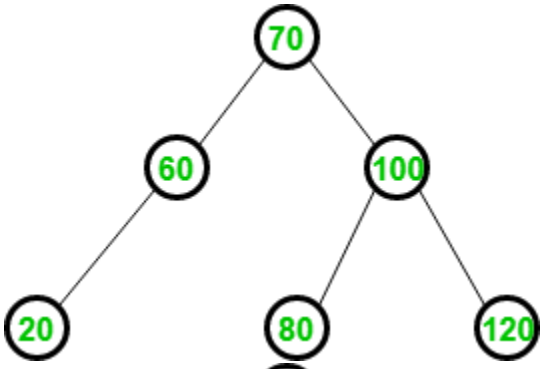
```
#include <stdio.h>
int jumble(int x, int y){
        x=2*x+y;
        return x;
}
int main(){
        int x=2, y=5;
        y=jumble(y,x);
        x=jumble(y,x);
        printf("%d \n", x);
        return 0;
}
```

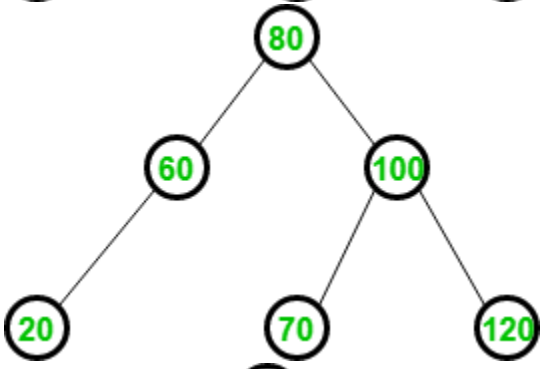
The value printed by the program is __26___

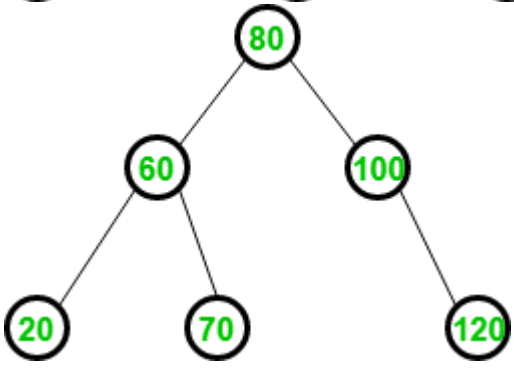
118. Consider the following AVL tree.



Which of the following is updated AVL tree after insertion of 70?

A.


B.


C.


D. None

119. What is the time, and space complexity of the following code:

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
   a = a + rand();
```

```
}
for (j = 0; j < M; j++) {
  b = b + rand();
}
```

A. O(N * M) time, O(1) space
B. O(N + M) time, O(N + M) space
C. O(N + M) time, O(1) space
D. O(N * M) time, O(N + M) space

120. What is the time complexity of the following code:

```
int a = 0;
for (i = 0; i < N; i++) {
  for (j = N; j > i; j--) {
    a = a + i + j;
  }
}
```

A. O(N)
B. O(N*log(N))
C. O(N * Sqrt(N))
D. O(N*N)

121. What is the time complexity of the following code:

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
  for (j = 2; j <= n; j = j * 2) {
    k = k + n / 2;
  }
}
```

A. O(n)
B. O(N log N)
C. O(n^2)
D. O(n^2Logn)

122. Rank the following functions by order of growth: $(n + 1)!$, $n!$, 4^n, $n \times 3^n$, $3^n + n^2 + 20n$, $(3/2)^n$, $n^2 + 200$, $20n + 500$, $2^{\lg n}$, $n^{2/3}$, 1.

123. Consider the following functions:

$f(n) = 2^n$
$g(n) = n!$
$h(n) = n^{\log n}$

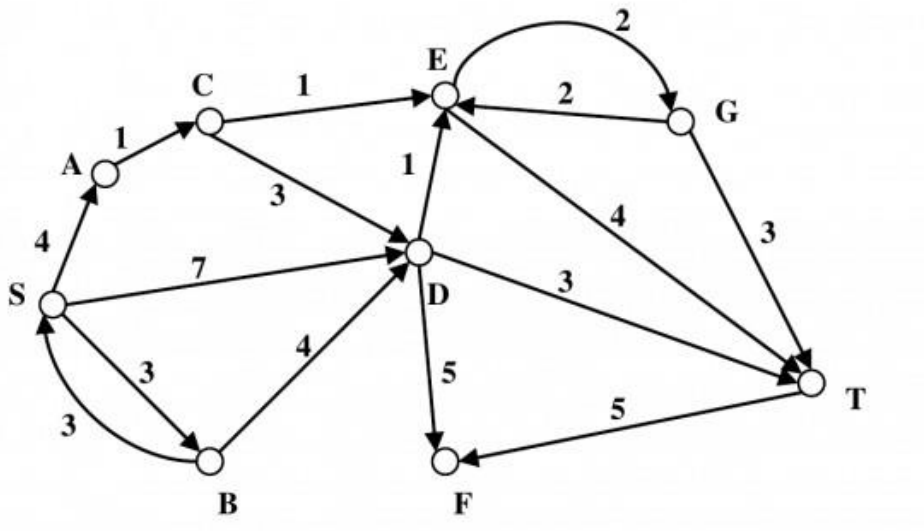
Which of the following statements about the asymptotic behavior of f(n), g(n), and h(n) is true?

A. f(n) = O(g(n)); g(n) = O(h(n))
B. f(n) = Ω (g(n)); g(n) = O(h(n))

C. g(n) = O(f(n)); h(n) = O(f(n))
D. h(n) = O(f(n)); g(n) = Ω (f(n))

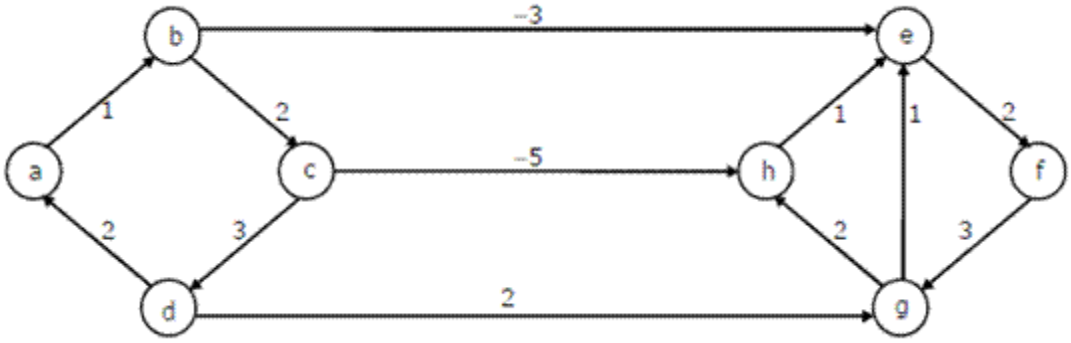
124. Which of the following three claims are correct?
I. (n + k)m = Θ(nm), where k and m are constants
II. 2n+1 = O(2n)
III. 22n+1 = O(2n)
A. I and II
B. I and III
C. II and III
D. I, II and III

125. Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered.



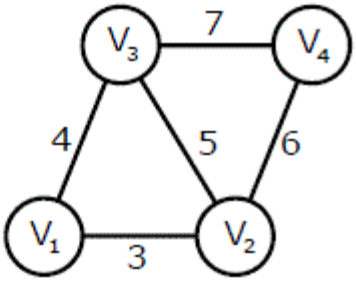
A. SDT
B. SBDT
C. SACDT
D. SACET

126. Dijkstra's single source shortest path algorithm when run from vertex a in the below graph, computes the correct shortest path distance to



A. only vertex a
B. only vertices a, e, f, g, h
C. only vertices a, b, c, d
D. all the vertices

127. An undirected graph G(V, E) contains n (n > 2) nodes named v1 , v2 ,....vn. Two nodes vi , vj are connected if and only if 0 < |i – j| <= 2. Each edge (vi, vj) is assigned a weight i + j. A sample graph with n = 4 is shown below. What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?



A. 1/12(11n^2 – 5n)
B. n^2 – n + 1
C. 6n – 11
D. 2n + 1

128. The length of the path from v5 to v6 in the MST of previous question with n = 10 is
A. 11
B. 25
C. 31
D. 41

129. Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry Wij in the matrix W below is the weight of the edge {i, j}. What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

A. 7
B. 8
C. 9
D. 10

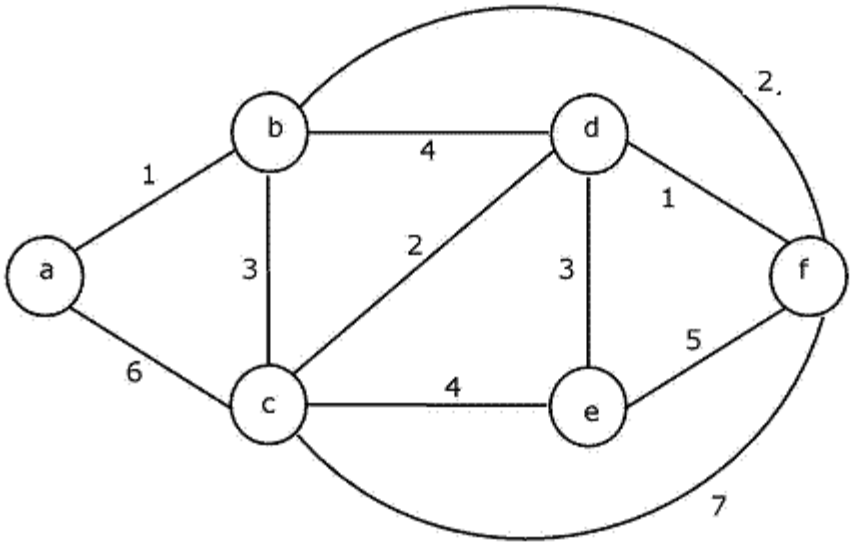
130. In the graph given in above question, what is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?

A. 7
B. 8
C. 9
D. 10

131. An undirected graph G has n nodes. Its adjacency matrix is given by an n × n square matrix whose (i) diagonal elements are 0's and (ii) non-diagonal elements are 1's. which one of the following is TRUE?

A. Graph G has no minimum spanning tree (MST)
B. Graph G has a unique MST of cost n-1
C. Graph G has multiple distinct MSTs, each of cost n-1
D. Graph G has multiple spanning trees of different costs

132. Consider the following graph:



Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using Kruskal's algorithm?

A. (a—b),(d—f),(b—f),(d—c),(d—e)
B. (a—b),(d—f),(d—c),(b—f),(d—e)
C. (d—f),(a—b),(d—c),(b—f),(d—e)
D. (d—f),(a—b),(b—f),(d—e),(d—c)

133. What is a data structure? Why do we need it? List some common data structures.

134. What is time and space complexity? Define Asymptotic notations with types.

135. How data structures are classified?

136. Differentiate linear and non-linear data structure.

137. Define ADT (Abstract Data Type). What are benefits of ADT?

138. What is the difference between an array and a linked list?

139. What are the ways of implementing linked list? What are the types of linked lists?

140. What is stack? What are the operations of the stack?

141. What are the applications of stack?

142. What are the methods to implement stack in C?

143. What is queue? What are the operations of the queue? What are the types of queue?

144. What are the methods to implement queue in C?

145. What are the applications of queue?

146. Difference between stack and queue?

147. Difference between tree and graph?

148. What is binary tree, complete binary tree and binary search tree?

149. What are the different types of tree traversal? Explain.

150. Define AVL tree. What is a balance factor in AVL trees?

151. What is the length of the path in a tree?

152. Define a heap. How can it be used to represent a priority queue?

153. Name the different ways of representing a graph?

154. What are the two traversal strategies used in traversing a graph? Explain each in detail.

155. What is insertion sort? How many passes are required for the elements to be sorted ?

156. What is linear search and binary search? Explain differences.

157. Difference between merge sort and quick sort?

158. What is the need for hashing?

159. What is linear probing?

160. Construct the AVL Tree for the given Sequence 21, 26, 30, 9, 4, 14, 28, 18,15,10, 2, 3, 7 step by step?

161. Show the result of inserting 12, 10, 15, 4, 1, 17, 3, 13, and 8 into an initially empty B tree with M = 3 and L = 2. To maintain consistency in answers, please follow the following rules:
    a. You should split nodes whenever there is an overflow due to insertion; that is, do not use adoption.
    b. When splitting a leaf node due to insertion overflow, keep half (rounded up) in the left node and half (rounded down) in the right.

162. Show the result of deleting 12, 13, and 15 in the above B-tree.