Nitish Kumar Gupta
Course: GATE
Computer Science Engineering(CS)

HOME
MY TEST
BOOKMARKS
MY PROFILE
REPORTS
BUY PACKAGE
ASK AN EXPERT
OFFER
EXCLUSIVE OFFER FOR OTS STUDENTS ONLY ON BOOK PACKAGES

SINGLE SUBJECT : PROGRAMMING AND DATA STRUCTURES (GATE - 2019) - REPORTS

OVERALL ANALYSIS COMPARISON REPORT SOLUTION REPORT

ALL(33) CORRECT(8) INCORRECT(13) SKIPPED(12)

Q. 1

Consider the following statements:
S_1 : In a binary search tree, the insertion operation is commutative.
S_2 : In a binary search tree, the deletion operation is commutative.
Which of the following is true?

Have any Doubt ?

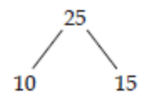
A S_1 is true, S_2 is true

B S_1 is true, S_2 is false

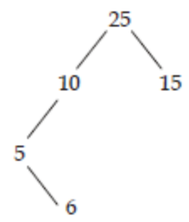
C S_1 is false, S_2 is true — Your answer is **Wrong**

D S_2 is false, S_2 is false — Correct Option

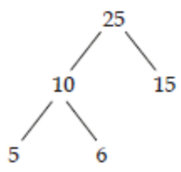
Solution :
(d)
Both are false statements
First consider a counter example for S_1.
Let the binary search tree be as follows:



Insert (5, 6) ⇒

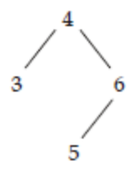


Now insert (6, 5) ⇒



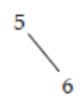
Both BSTs are different, hence S_1 is false.
Now consider statement S_2

Let BST be:



Delete (3, 4) ⇒

Delete (4, 3) ⇒

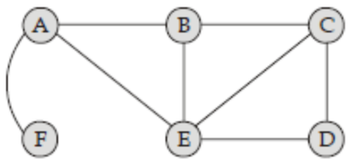


Again both BSTs are different, hence S_2 is also false.
Thus correct answer is (d).

QUESTION ANALYTICS +

Q. 2

Breadth First Search algorithm is executed on the graph given below. Which of the following is a correct order of visiting the nodes?



Have any Doubt ?

A ABCDEF

B BEADCF

C EABFCD

D DCEBAF — Your answer is **Correct**

Solution :
(d)
(a) is wrong, because if A is encountered first, B, E, F should be visited, but C is visited and hence this can't be the right sequence. Similarly we can check for other options and find that (d) is the correct sequence.

QUESTION ANALYTICS +

Q. 3

Consider an implementation of two stacks using a single array A[1 ..... N]. The 2 stacks grow from opposite ends of the array. Let top1 and top2 denote variables (top1 > top2) pointing to the location of the top most element in each of the two stacks. If space is to be used efficiently, the condition for stack overflow should be

Have any Doubt ?

A top1 + top2 = 1

B top1 + top2 = N

C top2 – top1 = –1 — Correct Option

Solution :
(c)
The condition for overflow will be

$$top2 = top1 - 1$$

Rearranging the equation, we get

$$top2 - top1 = -1$$

Hence answer (c) is correct.

D None of these — Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 4

Let T_1 and T_2 be the worst case running times to search for an element in a binary search tree with n-elements and a balanced binary search tree with $n^2\ 2^n$ elements. Then $T_1 + T_2$ will be equal to

Have any Doubt ?

A $O(2^n)$ — Your answer is **Wrong**

B $O(n^2)$

C O(n) — Correct Option

Solution :
(c)

$$T_1 = O(n)$$
$$T_2 = O(\log\ (n^2 2^n)$$
$$= O[\log n^2 + \log 2^n]$$
$$= O[2 \log n + n]$$
$$= O(n)$$

Hence $T_1 + T_2 = O(n)$

D $O(n.2^n)$

QUESTION ANALYTICS +

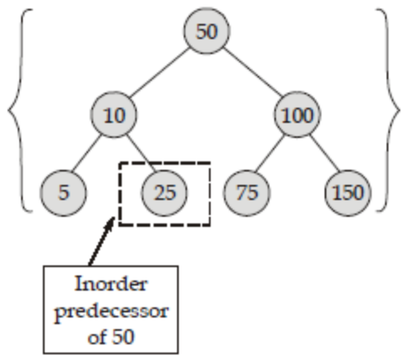
Q. 5

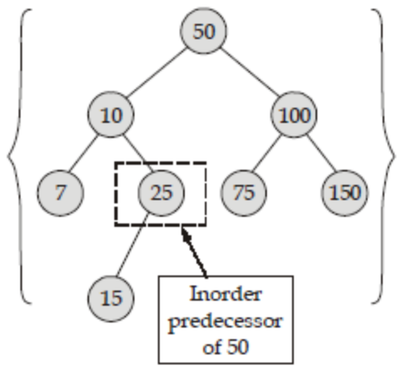
In delete operation of a binary search tree, we need inorder predecessor or successor of a node to be deleted where it has both left and right child. Which of the following is true about inorder predecessor in delete operation?

Have any Doubt ?

A Inorder predecessor is always either a leaf node, or a node with empty right child. — Your answer is **Correct**

Solution :
(a)
Predecessor of a node is always the maximum (largest) element of its left subtree.





B Inorder predecessor is always either a leaf node or a node with empty left child.

C Inorder predecessor is always a leaf node.

D Inorder predecessor is always an ancestor of the leaf node.

QUESTION ANALYTICS +

Q. 6

Let R denotes the class of recursive programs and I denotes the class of iterative programs. Which of the following is incorrect?

Have any Doubt ?

A Some programs belonging to class R don't terminate sometimes. — Your answer is **Wrong**

B For every program belonging to class I, there exists an equivalent program belonging to class R.

C Every program in K uses strictly more stack space compared to its equivalent program in J. Correct Option

Solution :
(c)
If a recursive program has non tail recursion in it, then stack space cannot be reduced. This is possible only which a program has tail recursion — its iterative equivalent can be freed from the use of the stack. Hence (c) is the required choice.

D None of these

QUESTION ANALYTICS +

Q. 7

Consider the following C declaration:
char (* (* x ( )) [ ]) ( );
Which of the following correctly describes x?

FAQ | Solution Video | Have any Doubt ?

A x is a function which returns pointer to an array of pointers to characters.

B x is a function which returns pointer to an array of pointers to function which returns char. Your answer is **Correct**

Solution :
(b)
Choices a, c, d are easily ruled out. B is the most appropriate choice.

C x is a pointer to a function which returns pointer to an array of pointers to function which returns characters.

D None of these

QUESTION ANALYTICS +

Q. 8

Consider 4 integer variables a, α, β, Γ. Consider the following expression.

$$a = (\alpha > \beta) \ ? \ ((\alpha > \Gamma) \ ? \ \alpha : \Gamma) : ((\beta > \Gamma) \ ? \ \beta : \Gamma)$$

Choose the correct combination of α, β and Γ so that the variable a gets the value 4 in the above expression.

FAQ | Have any Doubt ?

A 3, 4, 2 Your answer is **Correct**

Solution :
(a)
By inspection, we see that option (a) is correct.
Step 1:

$$\begin{pmatrix} 3 > 4 \\ \alpha > \beta \end{pmatrix} \Rightarrow \text{False}$$

Step 2:

$$\begin{pmatrix} 4 > 2 \\ \beta > \Gamma \end{pmatrix} \Rightarrow \text{True}$$

Hence the variable a gets the value of β, which is 4.
Hence (a) is the right choice.

B 6, 5, 3

C 5, 4, 5

D None of these

QUESTION ANALYTICS +

Q. 9

Consider a string str, declared as "MADEEASY". Assuming that the size of a character is 1 bytes, choose the most appropriate choice.

Have any Doubt ?

A Both strlen(str) and sizeof(str) output the value 8. Your answer is **Wrong**

B Strlen(str) outputs 8, but sizeof(str) outputs the value 9. Correct Option

Solution :
(b)
Strlen(str) counts the length of the string; thus strlen( ) excludes the null character, unlike sizeof( ) operator, which includes the null character.
Hence, the correct choice is (b).

C Strlen(str) outputs 9, but sizeof(str) outputs the value 8.

D Both strlen(str) and sizeof(str) output the value 9.

QUESTION ANALYTICS +

Q. 10

Which of the following is false regarding the implementation of queue using 2 stacks?

Have any Doubt ?

A If inserting an element takes O(1) time, then deletion must take O(n) time. Your answer is **Wrong**

B If inserting an element takes O(n) time, then deletion must take O(1) time.

C Both insertion and deletion will take O(1) time. Correct Option

Solution :
(c)
One and only one out of insertion and deletion will take O(n) time. Hence (c) is wrong.

D None of these

QUESTION ANALYTICS +

Q. 11

Let A be the result when the postfix expression below in evaluated.
6 2 3 + – 3 8 2 / + * 3 ^ 3 +
Let Y be the result of the following expression.
2 A * 16 +
Then the value of $\sqrt{Y}$ will be _____. (Upto 1 decimal place)

FAQ | Have any Doubt ?

26.6 (25.0 - 27.0) 26.6
Your answer is **Correct**

Solution :
26.6 (25.0 - 27.0)
A can be found by evaluating the postfix expression.
$\therefore \quad A = 346$
Then to get Y, substitute the value of A in the postfix expression and evaluate the same.
$\therefore \quad Y = 2 \times 346 + 16 = 708$
Now, $\sqrt{Y} = 26.6$

QUESTION ANALYTICS +

Q. 12

Consider a 2 dimensional array A[40 ..... 95, 40 ..... 95] in lower triangular matrix representation. The size of each element in the array is 1 byte. If the array is implemented in the memory in the form of row major order and base address of the array is 1000, the address of A[66] [50] will be _____.

FAQ | Have any Doubt ?

1361 Correct Option

Solution :
1361

$$\text{Location (A[66] [50])} = 1000 + \frac{(66-40)(66-40+1)}{2} + (50-40)$$
$$= 1361$$

QUESTION ANALYTICS +

Q. 13

Consider the following C code snippet:

```
main ( )
{
    int S[6] = {128, 256, 512, 1024, 2048, 4096};
    int *x = (int *) (& S + 1);
    printf ("%d", x);
}
```

Let the size of int is 4 bytes; the array starts from 2000 onwards. Then the o/p generated by the above code is _____.

Have any Doubt ?

2024 Correct Option

Solution :
2024

128	256	512	1024	2048	4096
2000	2004	2008	2012	2016	2020

x | 2024 |

= (int *) (& S + 1);
= (int *) (Base address of S + 1 * size of (S))
= 2000 + (24 bytes) * 1
⇓
(6 ints, 4 bytes each)
= 2024

QUESTION ANALYTICS +

Q. 14

Consider the given function magic ( ):

```
int magic (int n)
{
    static int r = 5;
    if (n <= 0) return 10;
    if (n > 3)
{
    r = 50;
```

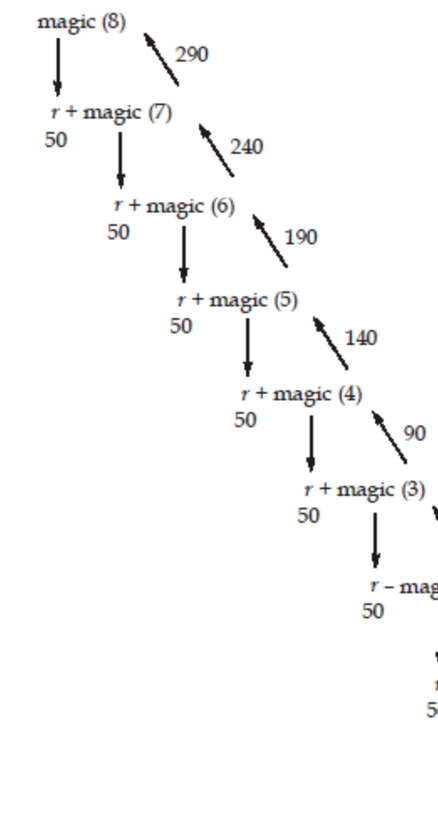
```
    return (r + magic (n – 1));
}
    return (r – magic (n – 1));
}
```

The output corresponding to the function call magic (8) is _____.

Have any Doubt ?

290 — Your answer is **Correct**290

Solution :
290



QUESTION ANALYTICS +

Q. 15

Consider the following C code:

```
omega (int *a, int n)
{
    if (n < = 0) return 0;
    else if (*a %3 = = 0)
        return (*a + f(a + 1, n –1));
    else return (*a – f(a + 1, n –1));
}
main ( ) {
    int a[ ] = {–12, 7, 13, –4, –11, 6};
    printf ("%d", omega (a, 6));
}
```

The output of this program will be _____.

Have any Doubt ?

-5 — Correct Option

Solution :
-5

The output will be = –12 + (7 – (13 – (– 4 – (– 11 – 6))))
= –5

Your Answer is -49

QUESTION ANALYTICS +

Q. 16

Consider a hash table of size 10 that employs open addressing with linear probing. The hash function is given by $h(k)$ = k mod 10. The hash table contains bins indexed from 0 to 9. A sequence of records with keys given below is inserted into an initially empty hash table.
83, 84, 95, 74, 23, 86, 41, 62, 72
The number of unsuccessful probes required to find the index of bin which contains the last element is _____.

Have any Doubt ?

7 — Correct Option

Solution :
7

	41	62	83	84	95	74	23	86	72
0	1	2	3	4	5	6	7	8	9

The last record inserted is 72.
So the search will start from 72 mod 10 = 2.
And since linear probing is used, we start probing sequentially and we continue this process till we encounter 72 at 9^{th} position. Hence, from position 2 to 8, all probes are unsuccessful.
∴ Number of unsuccessful probes = 8 – 2 + 1 = 7

Your Answer is 1

QUESTION ANALYTICS +

Q. 17

You're entrusted with the task of deleting a node in a singly linkedlist, whose data field is 'x'. Note that, the node which is to be deleted can be at any arbitrary position in the linked list. Consider the following scenarios.
S_1 : You're only provided with a pointer to the node which is to be deleted in the linked list.
S_2 : You're only provided with a pointer to the starting node of the linked list.
Which of the following options is correct?

Have any Doubt ?

A. In both the scenarios, deletion is possible for all inputs, and deletion will be more efficient in S_1 than S_2.

B. In both the scenarios, deletion is possible for all inputs, and deletion will be more efficient in S_2 than S_1.

C. In S_1, deletion is not possible in certain cases; but in S_2, deletion is possible for all inputs. — Correct Option

Solution :
(c)
(c) is the correct option, as in the first scenario, if the pointer is provided to the last node, then unless we have the starting address or address of second last node of the linked list, we cannot delete the last node. But any node can be deleted in the 2nd scenario.

D. In S_2, deletion is not possible in certain cases; but in S_1, deletion is possible for all cases.

QUESTION ANALYTICS +

Q. 18

Consider the following functions, googly( ), doosra( ) and teesra( ). Note that, a variable has a bool type if it holds a value in {true, false}. Also, $\log_2(n)$ computes the base 2 logarithm of the input number n.

```
FUNCTION 1
bool doosra(int n)
{
    return (ceil(log2(n)) == floor(log2(n)))
}
FUNCTION 2
bool googly(int n)
{
    if (n == 0) return false;
    while (n! = 1)
    {
    if (n% 2! = 0) return false;
    n = n/2;
    }
    return true;
}
FUNCTION 3
bool teesra(int x)
{
    return x && (! (x &(x - 1)));
}
```

Which of the above functions produce the same output for a given input?

Have any Doubt ?

A. Googly, Doosra

B. Doosra, Teesra

C. All 3 are equivalent — Correct Option

Solution :
(c)
All the 3 functions check if a given number is a power of 2.
Function 1: Checks if $\log_2(n)$ of a number is an integer. If yes, it returns true, else it returns false. So function 1 checks if a given number is a power of 2.
Function 2: The key here is that, a number which is a power of 2 has the bit pattern 10^* (1 followed by any number of zeroes). So at every step we keep checking if the number is even and keep dividing the number by 2 (right shift); if except for the most significant bit, a bit is found to be 1 (the number is odd at any point of time while right shifting), then the function 2 returns false. Else it returns true. So function 2 also checks if a given number is a power of 2.
Function 3: The observation is that, if a number n is power of 2, then $(n - 1)$ becomes the 1's complement of n. Hence function 3 also checks if a given number is a power of 2.

D. None of these

QUESTION ANALYTICS +

Q. 19

A queue is implemented using a singly linked list. The queue has a head pointer and a tail pointer, which point to the starting and the last node of the linked list respectively. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail. Further, let the number of nodes in the queue be n. Then the time complexity required by the most efficient algorithm for the implementation of enqueue( ) and dequeue( ) operation for this scenario will be:

FAQ | Have any Doubt ?

A. $O(n)$, $O(n)$

B. $O(n)$, $O(1)$

C. $O(1)$, $O(n)$ — Correct Option

Solution :
(c)
Even though insertion can easily be done in $O(1)$, deletion will take $O(n)$ time, since tail points to the last node, and in order to delete the node pointed by tail pointer, we need the address of the second last node for which we need to scan (traverse) the entire list, and thus (c) is the most appropriate choice.

D O(1), O(1)

QUESTION ANALYTICS +

Q. 20

Consider the following code snippet called 'Program X':

```
void f(int n)
{
    if (n <= 1) printf ("%d", n);
    else
    {
    f(n/3);
    printf("%d", n% 3);
    }
}
```

Which of the following implementations will produce the same output for $f(1023)$ as the above code?

Program P_1:

```
void f(int n)
{
    if (n/3) {
    f(n/3);
    }
    printf("%d", n% 3);
}
```

Program P_2:

```
void f(int n)
{
    if (n <= 1) printf("%d", n);
    else
    {
    printf("%d", n% 3);
    f(n/3);
    }
}
```

Have any Doubt ?

A Both P_1 and P_2

B Only P_1 Correct Option

Solution :
(b)
The program X prints the ternary equivalent of 1023. Program P_1 also prints the ternary equivalent of 1023. However, program P_2 prints the ternary equivalent of 1023 in reverse order.
Hence the answer is (b).

C Only P_2

D None of these Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 21

Consider a hypothetical machine which supports the following data types:
unsigned char: 1 Byte
unsigned short: 2 Bytes
int: 4 Bytes
Consider the following function red( )

```
int red(unsigned char a, unsigned short b)
{
    if (a = = 0) return b;
    else {
        a = a + 1;
        b = b * 2;
        return red(a, b); }
}
int main( ) {
    printf("%d", red((char) 240, 1));
    return 0;
}
```

What will be the output of the following program?

Have any Doubt ?

A The program terminates abnormally

B The program goes into infinite loop

C The program outputs 65536 (2^{16})

D None of these Correct Option

Solution :
(d)
The value returned by the above program will be zero. To understand this, here's the sequence of the recursive calls.

Output:

a = 240 and b = 1
a = 241 and b = 2
a = 242 and b = 4
a = 243 and b = 8
a = 244 and b = 16
a = 245 and b = 32
a = 246 and b = 64
a = 247 and b = 128
a = 248 and b = 256
a = 249 and b = 512
a = 250 and b = 1024
a = 251 and b = 2048
a = 252 and b = 4096
a = 253 and b = 8192

a = 254 and b = 16384
a = 255 and b = 32768
a = 0 and b = 0

The key here is that the range of unsigned char is [0 – 255] and that of unsigned short is [0 – 65535].

Hence when a = 255 and b = 32768, a + 1 should be 256 and b * 2 should be 65536; however due to overflow, a and b both become equal to zero as they are unsigned.

QUESTION ANALYTICS

Q. 22

Consider the following C code:

```
#include <stdio.h>
int sum(int A[ ], int n) {
int s = 0;
for (int i = 0; i < n; i++)
s+ =A[i];
return s; }
int main(void) {
int a[6] = {000, 001, 010, 011, 012, 100};
printf("%d", sum(a, 6));
return 0;
}
```

Let X be the output produced by the program. Let log(n) denote the logarithm of the given number n in base 2. Then log(X) will be equal to

Have any Doubt ?

A 6

B 7 — Your answer is **Correct**

Solution :
(b)

The catch here is that, some of the contents of the array are written in octal format. If a number is preceded by a zero, then the number is interpreted as an octal number in C. The code simply adds all the numbers up, and produces the output in decimal format.

Hence, the output will be:

(0 + 1 + 8 + 9 + 10 + 100) = 128
Thus log(128) = 7

C 8

D 9

QUESTION ANALYTICS

Q. 23

Let S be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence and then perform n pop operations. Assume that Push and Pop operations take X seconds each and Y seconds elapse between the end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S. The average stack-life of an element of this stack is

Have any Doubt ?

A n(X + Y)

B 3Y + 2X

C n(X + Y) – X — Correct Option

Solution :
(c)

Take n = 1. So it means that only 1 is pushed into the stack.
So time to push = X seconds
Now before 1 gets popped, another Y seconds will be elapsed as it is given that, the time between any two successive operations is Y seconds.
So time at this instant (before 1 gets popped out) = X + Y seconds
So now 1 will be popped out from the stack.
Therefore, lifetime of 1 = (X + Y) – X = Y seconds

Now in the options, put n = 1.

Option (a) becomes, X + Y	... wrong
Option (b) becomes, 3Y + 2X	... wrong
Option (d) becomes, Y + 2X	... wrong
Option (c) becomes, X + Y – X = Y	... correct

Hence the correct choice is (c).

D Y + 2X

QUESTION ANALYTICS

Q. 24

Consider the following 3 programs:

Program P_1 :

```
int *g(void) {
    int x = 10;
    return (&x);
}
```

Program P_2 :

```
int *g(void) {
    int *px;
    *px = 10;
```

```
    return px;
}
```

Program P_3 :

```
int *g(void) {
    int *px;
    px = (int*) malloc (sizeof (int));
    *px = 10; free(px);
    return px;
}
```

Which of the above three functions are likely to cause problems with pointers?

Have any Doubt ?

A Only P_2

B Only P_1 and P_3 — Your answer is **Wrong**

C Only P_1 and P_2

D P_1, P_2 and P_3 — Correct Option

Solution :
(d)
Since P_1 returns the address of a variable which is declared locally, P_1 may cause problems.
P_2 will cause a problem because px doesn't have any address and is being dereferenced.
P_3 also will cause problems because even though malloc has been used to allocate the memory into the heap, free( ) has been called and returning that address is simply asking for trouble.

QUESTION ANALYTICS +

Q. 25

Consider the following function, which takes the starting address of 2 linked lists as input and returns an integer in {0, 1}:

```
int foo (Node * head A, Node * head B)
{
    if (head A → data! = head B → data)
        return 0;
    else if (head A → data = = head B → data)
{
    if (head A → next && head B → next)
    return foo (head A → next, head B → next);
    else if (! head A → next && ! head B → next);
    return 1;
    else return 0;
}
```

The give function

Have any Doubt ?

A Returns 0 for all inputs

B Returns 1 for all inputs

C Return 1 if both linked lists are sorted in ascending order and returns 0 otherwise

D None of these — Correct Option

Solution :
(d)
The given function checks whether the given 2 linked lists are identical or not. If the linked lists are identical, it returns 1, otherwise 0 is returned.

QUESTION ANALYTICS +

Q. 26

Consider a hash table N slots. It is given that the collision resolution technique used is chaining. Assuming simple uniform hashing, what is the probability that the last k slots are unfilled after the first 'r' insertions?

Have any Doubt ?

A $\left(1-\frac{N}{k}\right)^r$

B $\left(1-\frac{k}{N}\right)^r$ — Correct Option

Solution :
(b)
Probability that last k slots are empty after first r iterations

$$= \frac{\overbrace{(N-k)(N-k)(N-k).....(N-k)}^{r \text{ times}}}{N^r}$$

$$= \frac{(N-k)^r}{N^r} = \left[\frac{N-k}{N}\right]^r$$

$$= \left(1-\frac{k}{N}\right)^r$$

C $\left(1+\frac{N}{k}\right)^{r-1}$

D $\left(1-\frac{k}{N}\right)^{r-1}$

QUESTION ANALYTICS

Q. 27

Consider the following function foobar( ), which takes a binary tree as input:

```
int foobar (struct node * root)
{
    if (! root) return 0;
    if (! root → left && ! root → right) return 10;
    else
    {
        int i = foobar (root → left);
        int j = foobar (root → right);
        return (i + j);
    }
}
```

What does the above function foobar compute?

Have any Doubt ?

A Sum of internal nodes of the binary tree

B Number of leaves of the binary tree

C Sum of leaves of the binary tree

D None of these — Your answer is **Correct**

Solution :
(d)
The above function returns 10 times the number of leaf nodes.
Hence the answer is (d).

QUESTION ANALYTICS

Q. 28

What is the time and space complexity required to check (by the most efficient algorithms) whether a given linked list is palindrome or not?

Have any Doubt ?

A $O(n)$, $O(n)$

B $O(n)$, $O(1)$ — Correct Option

Solution :
(b)
$O(n)$ time and $O(1)$ space
Algorithm:

- First find middle element of linked list.
- Now using middle position, break the linked list into 2 parts.
- Reverse the linked list so obtained in second half.
- Now check element-by-element, if both the halves so obtained are equal or not. If equal, the given linkedlist is palindrome otherwise not.

Time complexity → $O(n)$
Space complexity → $O(1)$

C $O(n^2)$, $O(1)$

D $O(\sqrt{n})$, $O(1)$

QUESTION ANALYTICS

Q. 29

Consider the following code:

```
#include <stdio.h>
#include <string.h>
int main ( )
{
    char *a = "MADEEASY";
    char *b = "GATECSIT2019";
    char *r = a;
    char *s = b;
    printf("%d", (int) strlen(b + 3[r] – 1[s]));
    return 0;
}
```

The output of the following program will be _____.

Have any Doubt ?

8 — Correct Option

Solution :
8

$$3[r] \equiv r[3] = \text{'E'}$$

$1[s] \equiv s[1] = 'A'$

Let ASCII value of A = x

Hence $(b + 3[r] - 1[s]) = (b + 4)$

Hence strlen $(b + 4) \Rightarrow 8$

Hence 8 is the answer.

Your Answer is 6

QUESTION ANALYTICS

Q. 30

Consider the following program:

```
void MadeEasy (int n)
{
    printf("*");
    if (n > 1)
    {
        MadeEasy (n/4);
        MadeEasy (n/4);
        MadeEasy (n/4);
        MadeEasy (n/4);
    }
}
```

Let X be the number of asterisks printed by the above function when $n = 1024$. Then the value of X will be _____. (Hint: Take n as a power of 4)

Have any Doubt ?

1365 — Correct Option

Solution :

1365

The value of X i.e. number of stars printed can be represented by the following recurrence.

$$\text{Number of stars } (n) = \begin{cases} 1 + 4 \cdot \text{Number of stars } (n/4); & n > 1 \\ 1; & \text{otherwise} \end{cases}$$

Taking n as power of 4[$n = 4^k$]

Number of stars $(4^k) = 1 + 4$ number of stars (4^{k-1})

Solving the recurrence, we get

$$X = \left[\frac{4^{k+1} - 1}{3}\right]$$

Now since $1024 = 2^{10} = 4^5$, put $k = 5$ in the above expression to get,

$$X = \frac{4^{5+1} - 1}{3} = 1365$$

QUESTION ANALYTICS

Q. 31

Consider the integer arrayA[1 ..... 100, 1 ..... 100] in which the elements are stored in Z representation. An example of a 5 × 5 array in Z representation is shown below:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ & & & a_{24} & \\ & & a_{33} & & \\ & a_{42} & & & \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \end{array}$$

If the base address of A is starting from 1000 onwards, size of each element is 1 bytes and A is stored in Row Major Order, then the address corresponding to A[100] [55] is _____.

Have any Doubt ?

1252 — Correct Option

Solution :

1252

$$\text{Loc}(A[100]\ [55]) = [100 + (99 - 2 + 1)1 + (55 - 1) + 1000]$$

- 100 elements in first row
- 1 elements in all rows except 1st and last
- Base address

$$= 1252$$

QUESTION ANALYTICS

Q. 32

Let X be the maximum possible height of A binary search tree with 54 nodes. Let Y be the maximum possible height of an AVL tree with 54 nodes. Given that height of a tree is defined as the longest path from root to a leaf node, the value of [X – Y] is _____.

FAQ | Have any Doubt ?

46 — Correct Option

Solution :

46

It's easy to see that X = (54 – 1) = 53 (skew tree)

For Y, use the recurrence relation,

$$N(h) = N(h-1) + N(h-2) + 1;\ h \geq 2$$
$$= 1 \text{ if } h = 0;\ 2 \text{ if } h = 1$$

Here $N(h)$ represents minimum number of nodes in AVL tree of height 'h'.

So if $h = 7$, we get

$$N(7) = 33 + 20 + 1 = 54$$

Hence Y = 7

Now the required value (X – Y) = 53 – 7 = 46

Your Answer is 48.22

QUESTION ANALYTICS

Q. 33

How many of the following statements is incorrect regarding the time complexity of binary search tree, AVL tree, min heap tree and binary tree?

(a) In BST, the time taken to find a non existent element 'x' in the best case is O(1).

(b) In AVL, the time taken to find a non existent element 'x' in the best case is O(logn).

(c) In Binary Tree, the time taken to find a non existent element 'x' in the best case is O(n).

(d) In Minheap Tree, the time taken to find a non existent element in the best case is O(n).

FAQ | Have any Doubt ?

1

Correct Option

Solution :

1

In min heap, we know that root is the smallest element. If in a search query, there's an element which is smaller than the root, we can simply discard it as 'non existent' element. Hence (d) is the only incorrect statement.

Your Answer is 2

QUESTION ANALYTICS