Branch : CSE/IT **Batch : Hinglish**

WEEKLY TEST – 02

Subject : Algorithms

Topic : Design Strategies

Maximum Marks 15

Q.1 to 5 Carry ONE Mark Each

[MCQ]

1. What is the tightest Big -O time complexity for the following code?

```
int fun(int n)
{
        if (n < = 0)
         {
          return 0;
         }
    else if (n> = 2000)
         {
          return 1;
         }
    int count = 0
    for (int a = 0; a <n; a++)
        {
        count ++;
        }
    fun (n/2);
    fun(n/2);
    return count;
}
```

(a) O(1) (b) O(n)
(c) O(n^2) (d) O(nlogn)

[MCQ]

2. What is the time complexity of an efficient algorithm to determine if an integer array x contains two consecutive integers?

(a) O(n) if x is sorted, (or) O(nlogn) otherwise
(b) O(nlogn) whether x is sorted or not sorted
(c) O(n) if x is sorted (or) O(n^2) otherwise
(d) O(n^2) whether or not x is sorted.

[MSQ]

3. Consider the following statements

(a) Inserting, Deleting in arrays takes O(N) time, containing N elements.
(b) Inserting, deleting in string takes O(N) time.
(c) Space complexity for each operation in a linked list is O(1)
(d) The worst-case time complexity of binary search tree is O(n).

[MCQ]

4. What will be the worst case time complexity for the following code?

```
int foo(int a[ ], int x)
 {
      int result = 0;
      for (int p = 0; p < x; p++){
        for (int q = 0; q < 7; q++){
          for (int r = 0; r < x; r++){
            {
               result + = q * q + a[r];
            }
      }
  }
 return result;
 }
```

(a) O(n) (b) O(n^2)
(c) O(nlogn) (d) None of the above

[MCQ]

5. Consider an stack of having 'n' elements, inorder to reverse the elements in stack first pop off the elements one by one from stack and enqueue them into a queue. Then deque the elements one by one from the queue and push them back onto the stack, then what is the time complexity of above operation?

(a) O(n) (b) O(n^2)
(c) O(nlogn) (d) O(logn)

Q.6 to 10 Carry TWO Mark Each

[MCQ]

6. A = {a_1, a_2 …… a_n} is an array consisting of 'n' distinct integers and 'a' is an integer, design an efficient algorithm to determine whether there are two elements in 'A' whose sum is exactly 'a'. Find the time complexity of the algorithm which was designed.

(a) O(nlogn) (b) O(n^2)
(c) O(n) (d) none of these

[MCQ]

7. Consider the following function foo()____?

```
void foo(int x)
    {
        int a, b, c, d;
        for (a = 0; a < 1000; a++)
          {
              for (b = 0 b< n; b++)
                  {
                      for (c = 0; c < b; c++)
                          printf("%d",c)
                  }
          }
    }
```

What is the worst case running time of the function foo() for any positive value of x?

(a) O(n) (b) O(n^2)
(c) O(n^3) (d) O(1)

[MCQ]

8. What is the running time of the following c-program. Assume that n ≥ 1 and time complexity as a function of n.

```
a = n;
while (a > 0)
    {
        b = n;
        while (b > 0)
          {
            Add = Add + 10;
               b = b/2
          }
        a = a – 10;
    }
```

(a) O(n^2) (b) O(n)
(c) O($n\log_2 n$) (d) O($n^2\log n$)

[MCQ]

9. Consider the following Snippet of code.

```
 Addn = 0;
for (a = 1 to n)
     for (b = a to 2n)
          Addn = Addn + 1;
```

What is the time complexity of above given code?

(a) O(nlogn) (b) O(n)
(c) O(n^2) (d) O($n^2\log n$)

[MCQ]

10. Consider T(n + 1) = T(n) + 4n and T(n) = 1 then what is the value of T(n) in terms of n?

(a) $n^2 – n + 1$ (b) $2n^2 – 2n + 1$
(c) $2n^2 – 2n + 1$ (d) None of the above

Answer Key

1. (a)
2. (a)
3. (a, b, c, d)
4. (b)
5. (a)
6. (a)
7. (b)
8. (c)
9. (c)
10. (b)

Hints and Solutions

1. (a)

When n < = 0 which will return 0.

$\therefore$ O(1)

When n > = 2000, returns 1

$\therefore$ O(1)

When O < n < 2000, Here loop will run for a constnat time. Hence time complexity for large value of n is O(1) and the complexity of above code is O(1)

$\therefore$ option (a) is correct.

2. (a)

In order to check consecutive integer in a sorted array x, check

```
for (a = 0; a ≤ n; a++){
        if(x[a] = = x[a + 1] – 1)
            {
                print("exixts");
                Break
            }
}
```

This can be done in O(n) for sorted array.

for uinserted array, firstly sort the array in O(nlogn) time and then apply above algorithm.

$\therefore$ correct option is (a).

3. (a, b, c, d)

(a) yes, Inserting, Deleting in arrays takes O(N) time. Containing N elements

(b) yes, Inserting, deleting in string takes O(N) time, containing N-elements.

(c) yes, Space complexity for each operation in a linked list O(1)

(d) yes, The worst case time complexity of binary search tree is O(n).

4. (b)

p runs for n times(x)

q runs for 7 times

r runs for n times (x)

$\therefore 7n^2 \Rightarrow O(n^2)$

5. (a)

Popping off all elements from stack and enqueuing them into queue will take O(n) time. Similarly enqueuing all n elements from queue and push them onto stack, again it will take O(n)

Total time = O(n + n)

= O(2n)

= O(n)

6. (a)

Firstyly sort the given array 'A' in O(nlogn) time, after sorting take two pointers and start them from the left and right extreme of the sorted array 'A' and move them inwords. i.e… the left pointer moving right and the right pointer morining left. Check the sum of two number pointed by these pointers. If the sum is equal to x, then stop and return it otherwise continue above process till the pointers meet. This process takes time complexithy of O(n)., So the total time taken is O(nlogn).

$\therefore$ option (a) is correct.

7. (b)

$$\text{foo}(n)\sum_{a=0}^{999}\sum_{b=0}^{n-1}\left(\sum_{c=0}^{b-1}1\right)=O(n^2)$$

8. (c)

As we can see that, inner loop will be executing $\log_2 n$ time for every a.

$$b=n,\frac{n}{2},\frac{n}{4}......\frac{n}{n} \Rightarrow \log_2 n \text{ series}$$

outerloop: a = n, n – 10, n – 20 ….. 1

$$\therefore\ O\left[\frac{n}{10}\right]=O(n)$$

$\therefore$ Time complexity (T.C) = $O(\log_2 n)$

Hence option (c) is correct.

9. (c)

$$\sum_{a=1}^{n}\sum_{b=1}^{2n}(1)=\sum_{a=1}^{n}\left(1+1+1+1+....(2n-a+1)\text{times}\right)$$

$$= \sum_{a=1}^{n}(2n-a+1)=2n\sum_{a=1}^{n}(1)-\sum_{a=1}^{n}a+\sum_{a=1}^{n}(1)$$

$$= 2n(n)-\frac{n(n+1)}{2}+n$$

$= O(n^2)$

$\therefore$ option (c) is correct.

10. (b)

Given

$T(n + 1) = T(n) + 4(n)$

$T(n) = T(n - 1) + 4\ (n - 1)$

$T(n) = T(n - 2) + 4\ (n - 2) + 4\ (n - 1)$

$T(n) = T(n - 3) + 4\ (n - 3) + 4\ (n - 2) + 4(n - 1)$

$= T(n - k) + 4(n - k) + 4\ (n - k + 1) + \ldots\ldots$

$4\ (n - 3) + 4\ (n - 2) + 4\ (n - 1)$

let $n - k = 1 \Rightarrow k = n - 1$

$T(n) = T(1) + 4\ (1) + 4(2) + 4(3) + \ldots..\ 4\ (n - 2) + 4\ (n - 1)$

$= 1 + 4\ \{1 + 2 + 3 + \ldots\ldots\ n - 1\}$

$= 1 + \dfrac{4n(n-1)}{2} \Rightarrow 1 + 2n(n - 1)$

$= 2n^2 - 2n + 1$

$\therefore$ option (b) is correct.