MADE EASY PUBLICATIONS

POSTAL BOOK PACKAGE 2025

CONTENTS

COMPUTER SCIENCE & IT

Objective Practice Sets

Algorithms

1 CHAPTER Asymptotic Analysis of Algorithms

Multiple Choice Questions

Q.1 The concept of order (Big O) is important because
(a) It can be used to decide the best algorithm that solves a given problem
(b) It determines the maximum size of a problem that can be solved in a given amount of time
(c) It is the lower bound of the growth rate of algorithm
(d) Both (a) and (b) above

Q.2 Let $f(n) = \Omega(n)$ and $g(n) = \Omega(n^2)$. Then $f(n) + g(n)$ is
(a) $\Omega(n)$ (b) $\theta(n)$
(c) $\Omega(n^2)$ (d) $O(n)$

Q.3 The order of an algorithm that finds whether a given Boolean function of 'n' variables, produces a 1 is
(a) Constant (b) Linear
(c) Logarithmic (d) Exponential

Q.4 $f(n) = 5n^2 + 6n + 10$
Which will be the exact value for $f(n)$?
(a) $\theta(n^2)$ (b) $O(n^2)$
(c) $o(n^2)$ (d) $\Omega(n^2)$

Q.5 Match the following groups:

	Group-I ($n > 0$)		Group-II
A.	$3n + 4n^2 + 5n \log n$	1.	$O(1)$
B.	$n + \log n + \log \log n$	2.	$O(\log n)$
C.	$10 + n + n \log n + \log n$	3.	$O(n)$
D.	$10 + 10000 + 100000$	4.	$O(n \log n)$
		5.	$O(n^2)$

Codes:

	A	B	C	D
(a)	5	3	4	2
(b)	5	4	3	1
(c)	5	3	4	1
(d)	5	4	3	2

Q.6 Let $f(n) = \Omega(n)$ and $g(n) = \Omega(n^2)$. Then $f(n) + g(n)$ is
(a) $\Omega(n)$ (b) $\theta(n)$
(c) $\Omega(n^2)$ (d) $O(n)$

Q.7 Find which of the following is not correct?
(a) $\sum_{i=1}^{n} \sqrt{i} = O(n^{3/2})$
(b) $n^2 \log n = \Theta(n^2)$
(c) $100n^3 + 2n^2 = \Omega(n^2)$
(d) $n! = O(n^n)$

Q.8 Unrestricted use of Goto is harmful, because it
(a) increase running time of programs.
(b) makes debugging difficult.
(c) results in the compiler generating longer machine code.
(d) increase memory requirement of programs.

Q.9 Each of the function $2^{\sqrt{n}}$ and $n^{\log n}$ has a growth rate .... than that of any polynomial.
(a) Greater (b) Less
(c) Equal to (d) Uncertain

Q.10 Consider the following function:

```
void f(int n)
{
    int i, j, k, m;
    for (i = 0; i < 100; i ++)
    {
        for (j = 0; j < n; j++)
        {
            for (k = 0; k < j; k++)
                printf("%d", k);
        }
    }
}
```

What is the worst case running time of the function f for any positive value of n?
(a) $O(1)$ (b) $O(n)$
(c) $O(n^2)$ (d) $O(n^3)$

Q.11 Consider the following function:

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

The return value of the function is

(a) $\Theta(n^2)$ (b) $\Theta(n^2 \log n)$
(c) $\Theta(n^3)$ (d) $\Theta(n^3 \log n)$

Q.12 Consider the following segment of c-code:

```
int j, n;
j = 1;
while (j < = n)
```

The number of comparisons mode in the execution of the loop for any $n > 0$ is

(a) $\lceil \log_2 n \rceil + 1$ (b) n
(c) $\lceil \log_2 n \rceil$ (d) $\lfloor \log_2 n \rfloor + 1$

Q.13 In the following C function, let $n \geq m$.

```
int gcd (n, m)
{   if (n% m = = 0} return m;
    n = n% m;
    return gcd (m, n);
}
```

How many recursive calls are made by this function?

(a) $\Theta(\log_2 n)$ (b) $\Omega(n)$
(c) $\Theta(\log_2 \log_2 n)$ (d) $\Theta\left(\sqrt{n}\right)$

Q.14 What is time complexity of 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) $O(N * \text{Sqrt}(N))$ (d) $O(n^2)$

Q.15 What does it mean when we say that one algorithm X is asymptotically more efficient than Y?

(a) X will always be a better choice for small inputs.
(b) X will always be a better choice for large inputs.
(c) Y will always be a better choice for small inputs.
(d) X will always be a better choice for all inputs.

Q.16 What is time complexity of following code:

```
int a = 0, i = N;
while (i > 0) {
    a = a + i;
    i/ = 2;
}
```

(a) $O(N)$ (b) $O(\text{Sqrt}(N))$
(c) $O(N/2)$ (d) $O(\log N)$

Q.17 What is time complexity of fun( )?

```
int fun (int n)
{
    int count = 0;
    for (int i = n; i > 0; i/ = 2)
      for (int j = 0; j < i; j++)
      count + = 1;
    return count;
:
```

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

Q.18 Consider the following recursive function:

```
int F (int array [ ], int n)
{
    int S = 0;
    if (n = = 0)
        return 0;
    S = F (array, n – 1)
    if (array [n – 1] < 0)
        S = S + 100;
    return S;
}
```

What is the worst case time complexity of the above function?

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

Q.19 What is time complexity of following program?

```
Void fun (int n){
int i, j, counter = 0;
for (i = 1; i ≤ n; i++) {
for (j = 1; j × j ≤ n; j++) count++;}}
```

(a) $O(n^2)$ (b) $O(n^{3/2})$
(c) $O(n \log n)$ (d) $O(n \log \log n)$

Q.20 Consider the following functions:

1. $n!$
2. a^n, a is constant, $a > 0$
3. n^n
4. n^k, k is a constant, $k > 0$
5. e^n

Choose the correct statement which ranks all functions by order of growth.

(a) $2 < 4 < 5 < 1 < 3$
(b) $4 < 2 < 3 < 5 < 1$
(c) $4 < 5 < 2 < 1 < 3$
(d) $4 < 2 < 5 < 1 < 3$

Q.21 Consider the following functions:

n, $\log n$, $\sqrt{n}$, $\log(\log n)$, $\dfrac{n}{\log n}$, $(\log n)^2$,

$\sqrt{n}\log n$, $n\log n$

Identify the functions in increasing order of growth.

(a) $\dfrac{n}{\log n}$, $\log(\log n)$, $(\log n)$, $(\log n)^2$, $\sqrt{n}$, $\sqrt{n}\cdot\log n$, n
(b) $\log(\log n)$, $\log n$, $(\log n)^2$, $\sqrt{n}$, $\sqrt{n}\log n$, $\dfrac{n}{\log n}$, n, $n \log n$
(c) $\log(\log n)$, $\log n$, $(\log n)^2$, $\sqrt{n}$, $\dfrac{n}{\log n}$, $\sqrt{n}\log n$, n, $n \log n$.
(d) None of these

Q.22 Which one of the following is true?

1. $an = o(n^2)$ (small oh) $a \geq 0$
2. $an^2 = O(n^2)$ (big oh) $a > 0$
3. $an^2 \neq o(n^2)$ (small oh) $a > 0$

(a) Only 1 and 2 are correct
(b) Only 1 is correct
(c) 1 and 3 are correct only
(d) All are correct

Q.23 Consider the following statements:

1. Any two functions f, g are always comparable under big-oh that is $f = O(g)$ or $g = O(f)$
2. If $f = O(g)$ and $f = O(h)$ then, $g(n) = \theta(h)$

Select correct option:

(a) 1 is true and 2 is false
(b) 1 is false and 2 is true
(c) Both are false
(d) Both are true

Q.24 1. $\frac{1}{2}n^2 = \omega(n)$, 2. $\frac{1}{2}n^2 = \omega(n^2)$

Which of the following is true?

(a) 1 is correct
(b) 2 is correct
(c) 1 and 2 both are correct
(d) None of these

Q.25 Consider the following functions:

$$f(n) = 2^{\log_2 n}$$
$$f(n) = n^{\log_2 n}$$
$$h(n) = n^{1/\log_2 n}$$

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

(a) $f(n) = \Omega(g(n))$ and $g(n) = O(h(n))$
(b) $g(n) = \Omega(h(n))$ and $f(n) = O(f(n))$
(c) $f(n) = O(g(n))$ and $g(n) = \Omega(h(n))$
(d) $g(n) = O(h(n))$ and $h(n) = O(g(n))$

Q.26 Suppose $f, g, h, k : N \rightarrow N$.

If $f = O(h)$ and $g = O(k)$, then

(a) $f + g = O(h + k)$
(b) $fg = (hk)$
(c) Both (a) and (b) above
(d) None of the above

Q.27 Let $g(n) = \Omega(n)$, $f(n) = O(n)$ and $h(n) = \theta(n)$ then what is the time complexity of $[g(n)\ \ f(n) + \ h(n)]$

(a) $O(n)$ (b) $\theta(n)$
(c) $\Omega(n)$ (d) $\theta(n^2)$

Q.28 Consider an array of n element with sorted order, if any element i appear more than half the number of element, what is the time complexity to count the number of occurrences of i?

(a) $O(\log n)$ (b) $O(1)$
(c) $O(n)$ (d) $O(\log\log n)$

Q.29 Find the time complexity of the following summation. Assume that k is a constant, $k > 0$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{1}{k}$$

(a) $O(n)$ (b) $O(n^2)$
(c) $O(n^3)$ (d) None of these

Answers — Asymptotic Analysis of Algorithms

1.	(d)	2.	(c)	3.	(d)	4.	(b)	5.	(c)	6.	(c)	7.	(b)	8.	(b)	9.	(a)
10.	(c)	11.	(b)	12.	(d)	13.	(a)	14.	(d)	15.	(b)	16.	(d)	17.	(b)	18.	(a)
19.	(b)	20.	(d)	21.	(b)	22.	(c)	23.	(c)	24.	(a)	25.	(c)	26.	(c)	27.	(c)
28.	(a)	29.	(b)	30.	(a)	31.	(d)	32.	(d)	33.	(d)	34.	(c)	35.	(a)	36.	(c)
37.	(a)	38.	(a)	39.	(a)	40.	(c)	41.	(b)	42.	(c)	43.	(b)	44.	(c)	45.	(a)
46.	(a)	47.	(d)	48.	(d)	49.	(d)	50.	(a)	51.	(a)	52.	(a)	53.	(c)	54.	(d)
55.	(b)	56.	(d)	57.	(a)	58.	(c)	59.	(d)	60.	(c)	61.	(b)	62.	(b)	63.	(a)
64.	(b)	65.	(d)	66.	(b)	67.	(b)	68.	(a)	69.	(c)	70.	(c)	71.	(c)	72.	(c)
73.	(c, d)		74.	(a, b)		75.	(a, b, d)		76.	(a, c)							

Explanations — Asymptotic Analysis of Algorithms

1. (d)

Big O notation gives worst case limit for a given problem. Also find out the least upper bound of problem.

2. (c)

Given

$$f(n) = \Omega(n)$$

i.e. $f(n) \geq c_1(n)$

$f(n)$ can be anything but atleast (n) not less than (n)

Given

$$g(n) = \Omega(n^2)$$

i.e. $g(n) \geq c_2(n^2)$

$g(n)$ can be anything but atleast (n^2) not less than (n^2)

$$f(n) + g(n) = \Omega((n) + (n^2)) = \Omega(n^2)$$

Here we can not comment about upper bound.

3. (d)

In the worst case it has to check all the 2^n possible input combinations, which is exponential.

4. (b)

$$f(n) = 5n^2 + 6n + 10$$

So, $\underset{\text{(Big-oh)}}{O(n^2)}$ is exact value of $f(n)$.

So, answer is (b).

5. (c)

$3n + 4n^2 + 5n \log n = O(n^2)$

$n + \log n + \log \log n = O(n)$

$10 + n + n \log n + \log n = O(n \log n)$

$10 + 10000 + 100000 = O(1)$

6. (c)

Given

$$f(n) = \Omega(n)$$

i.e. $f(n) \geq c_1(n)$

$f(n)$ can be anything but atleast (n) not less than (n)

Given

$$g(n) = \Omega(n^2)$$

i.e. $g(n) \geq c_2(n^2)$

$g(n)$ can be anything but atleast (n^2) not less than (n^2)

$$f(n) + g(n) = \Omega((n) + (n^2)) = \Omega(n^2)$$

Here we can not comment about upper bound.

7. (b)

$n^2 \log n \leq k.n^2$ will not satisfy for any constant k.

$\therefore$ Option (c) is not correct.

8. (b)

Unrestricted we of goto statement is harmful because it makes more difficult to verifying programs i.e., use of goto can results in unstructured code and there can be blocks with multiple entry and exit which can cause difficulty which debugging of program.

9. (a)

$2^{\sqrt{n}}$ and $n^{\log n}$ grows exponentially which have growth rate greater than any polynomial.

10. (c)

$$f(n) = \sum_{i=0}^{99} \sum_{j=0}^{n-1} \left(\sum_{k=0}^{j-1} 1 \right) = O(n^2)$$

11. (b)

Outer loop execute for $\frac{n}{2}+1$ iterations. Inner loop executes for $\log_2 n$ iterations. In every iteration of inner loop $\frac{n}{2}$ is added to k.

Return value = $\frac{n}{2}$ × number of outer loops × number of inner loops

$$= \frac{n}{2} \times \left(\frac{n}{2}+1\right)(\log n)$$

$$= O(n^2 \log n)$$

12. (d)

Let the increment of j is $2^0, 2^1, \ldots .. 2^i$ for some value of i so, according to the question for while loop; $2i \le n$ or $i \le \log_2 n$.

One extra comparison required for the termination of while loop.

So, total number of comparisons

$$= i + 1 = \lfloor \log_2 n \rfloor + 1.$$

13. (a)

Let, $T(m, n)$ be the total number of steps.

So, $T(m, 0) = 0$, $T(m, n) = T(n, m \bmod n)$ on average

$$T_n = \frac{1}{n} \sum_{0 \le k \le n} T(k, n)$$

$$T_n \approx 1 + \frac{1}{n}(T_0\ T_1 + \ldots + T_{n-1})$$

$$T_n \approx S_n$$

$$S_n = 1 + \frac{1}{n}(S_0\ S_1 + \ldots + S_{n-1})$$

$$S_n = 1 + \frac{1}{n+1}(S_0\ S_1 + \ldots + S_n)$$

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

$$= 1 + \frac{1}{n+1}$$

$$= S_n + \frac{1}{n+1}$$

So, $T_n \approx \Theta(\log_2 n) + 0(1)$

$T \approx \Theta(\log_2 n)$

14. (d)

The above code runs total number of times

$= N + (N-1) + (N-2) + \ldots .. + 1 + 0$

$= N * (N+1)/2$

$$= \frac{1}{2} * N \wedge 2 + \frac{1}{2} * N = O(N \wedge 2) \text{ times}$$

15. (b)

In asymptotic analysis, we consider growth of algorithm in terms of input size. An algorithm X is said to be asymptotically better than Y if X takes smaller time than Y for all input sizes n larger than a value n_0 where $n_0 > 0$.

16. (d)

We have to find smallest x such that $N/2$ ^ X N

$$X = \log(N)$$

So, $O(\log N)$ is time complexity.

17. (b)

For n time, inner loop will execute for n times.

For $\frac{n}{2}$ time, inner loop will execute for $\frac{n}{2}$ times.

For $\frac{n}{4}$ time, inner loop will execute for $\frac{n}{4}$ times

and do on ...

So, time complexity:

$$T(n) = O\left(n + \frac{n}{2} + \frac{n}{4} + \ldots + 1\right) = O(n)$$

18. (a)

Recurrence relation of function F

$F(n) = 0$ if $n = 0$

$F(n) = F(n-1) + 1,\ n > 0$

Time complexity = $O(n)$

19. (b)

for ($i = 1$; $i \le n$; i++) $\Rightarrow$ $O(n)$

for ($j = 1$; $j \times j \le n$; j++) $\Rightarrow$ $O(\sqrt{n})$

count ++;

Total time complexity = $O(n \times n^{1/2}) = O(n^{3/2})$

2 CHAPTER Recurrence Relations

Multiple Choice Questions & NAT Questions

Q.1 Let $T(n) = [n(\log(n^3) - \log n) + \log n]n + \log n$
Find complexity of $T(n)$.
(a) $O(n \log n)$ (b) $O(n^2)$
(c) $O(n^2 \log n)$ (d) $O(n^3)$

Q.2 Let T_n be the recurrence relation is defined as follows:

$$T_n = \begin{cases} 0, & n = 0 \\ T_{n-1} + n, & n \ge 1 \end{cases}$$

Find the value of T_n?
(a) n (b) n^2
(c) $\frac{n^2+n}{2}$ (d) $\frac{n^2-n}{2}$

Q.3 What is time complexity of recurrence equation

$$T(n) = T(\sqrt{n}) + 1$$

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

Q.4 What is complexity of recurrence equation
$T(n)$ = sqrt(2) + $(n/2)$ + sqrt (n)
(a) $\theta(n(\log n + 1))$ (b) $\theta(n^2(\log n + 1))$
(c) $\theta(\sqrt{n}(\log n + 1))$ (d) $\theta(n^3(\log n + 1))$

Q.5 $T(n) = T\left(\frac{2n}{3}\right) + 1$ then $T(n)$ is equal to
(a) $\Theta(\log_2 n)$ (b) $\Theta(n \log_2 n)$
(c) $\Theta(n^2)$ (d) $\Theta(n)$

Q.6 The running time of an algorithm is given by
$T(n) = T(n-1) + T(n-2) - T(n-3)$, if $n > 3$
n, otherwise
The order of this algorithms is
(a) $O(n)$ (b) $O(\log n)$
(c) $O(n^n)$ (d) $O(n^2)$

Q.7 What should be the relation between $T(1)$, $T(2)$ and $T(3)$, so that in above question gives an algorithm whose order is constant?
(a) $T(1) = T(2) = T(3)$ (b) $T(1) + T(3) = 2T(2)$
(c) $T(1) - T(3) = T(2)$ (d) $T(1) + T(2) = T(3)$

Q.8 What is complexity of recurrence eq.

$$T(n) = 2T(n/4) + \sqrt{3}$$

(a) $O(\sqrt{n})$ (b) $O(n)$
(c) $O(n^2)$ (d) $O(n \log n)$

Q.9 Which recurrence relation satisfy the sequence: 2, 3, 4, ..., for $n \ge 1$.
(a) $T(N) = 2T(N-1) - T(N-2)$
(b) $T(N) = 2T(N-1) + T(N-2)$
(c) $T(N) = N + 1$
(d) None of these

Q.10 Which one of the following correctly determines the solution of the recurrence relation with $T(1) = 1$?

$$T(n) = 2T\left(\frac{n}{2}\right) + \log n$$

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

Q.11 The time complexity of computing the transitive closure of a binary relation on a set of n elements is known to be
(a) $O(n \log n)$ (b) $O(n^{3/2})$
(c) $O(n^3)$ (d) $O(n)$

Q.12 Let $T(n) = T(n-1) + \frac{1}{n}$; $T(1) = 1$; Then $T(n) = ?$
(a) $O(n^2)$ (b) $O(n \log n)$
(c) $O(\log n)$ (d) $O(n^2 \log n)$

Q.13 A certain problem is having an algorithm with the following recurrence relation.

$$T(n) = T(\sqrt{n}) + 1$$

How much time would the algorithm take to solve the problem?
(a) $O(\log n)$ (b) $O(n \log n)$
(c) $O(\log\log n)$ (d) $O(n)$

Q.14 Consider the following recurrence relation:

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2 \log n,\ n > 1 \text{ and}$$

$T(1) = 1$

Find the time complexity of $T(n)$

(a) $O(n^2)$ (b) $O(n^2 \log n)$
(c) $O(n^3)$ (d) $O(n^3 \log n)$

Q.15 What is complexity of recurrence (n is a natural number)

$$T(n) = \begin{cases} 7T(n/3) + n^3; & n > 2 \\ 1; & n \le 2 \end{cases}$$

(a) $O(n)$ (b) $O(n^2)$
(c) $O(n^3)$ (d) $O(n \log n)$

Q.16 What is time complexity of the following recurrence relation and step to derive the same.

$$T(n) = T(\sqrt{n}) + \log(\log n)$$

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

Q.17 Let $T(n) = T(n/5) + T(7n/10) + c.n$, where c is a constant. Find running time of $T(n)$.

(a) $\theta(n)$ (b) $\theta(n \log n)$
(c) $\theta(n^2)$ (d) None of these

Q.18 Let $T(n) = T(n-1) + \frac{1}{n}$. Then $T(n)$ is

(a) $\Theta(1)$ (b) $\Theta(\log n)$
(c) $\Theta(\log \log n)$ (d) $\Theta(n)$

Q.19 Given $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$. Then

(a) $T(n) = \theta(n^3)$ (b) $T(n) = \theta(n^2 \log n)$
(c) $T(n) = \theta(n^2)$ (d) $T(n) = \theta(n^3 \log n)$

Q.20 What is the value of $T(n)$ for given recurrence relation

$$T(n) = T\left(\frac{n}{2}\right) + 2;\ T(1) = 1$$

when n is power of 2

(a) $2(\log n + 1)$ (b) $2 \log n$
(c) $\log n + 1$ (d) $2 \log n + 1$

Q.21 What is the time complexity of the following recurrence relation $T(k) = T\left(\frac{k}{2}\right) + T\left(\frac{k}{2}\right) + k^2$?

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

Q.22 Which of the following function given by their recurrences grows the fastest asymptotically?

(a) $T(n) = 8T\left(\frac{n}{4}\right) + 100n^2$

(b) $T(n) = 81T\left(\frac{n}{9}\right) + 10n^2$

(c) $T(n) = 16T\left(\frac{n}{4}\right) + 100(n \log n)^{1.99}$

(d) $T(n) = 100T\left(\frac{n}{100}\right) + n \log^2 n$

Q.23 Let $T(n)$ be the function defined by $T(1) = 1$,

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\sqrt{n} \text{ for } n \ge 2$$

Which of the following statements is true?

(a) $T(n) = O(\sqrt{n})$ (b) $T(n) = O(n)$
(c) $T(n) = O(\log n)$ (d) None of these

Q.24 Solve the recurrence relation to find:

$$T(n),\ T(n) = 4T\left(\frac{n}{2}\right) + n$$

(a) $\Theta(\log_2 n)$ (b) $\Theta(n^2)$
(c) $\Theta(n^2 \log_2 n)$ (d) None of these

Q.25 Solve $T(n) = 2T\left(\frac{n}{2}\right) + n^3$

(a) $\Theta(n^3)$ (b) $(n^3 \log_n n)$
(c) $\Theta(n \log n_2 n)$ (d) None of these

Q.26 What is complexity of following recurrence relation

$$T(n) = \sqrt{n}T(\sqrt{n}) + 100n$$

(a) $\theta(n \log \log n)$ (b) $\theta(n \log n \log n)$
(c) $\theta(n)$ (d) $\theta(n \log n)$

Q.27 What is complexity of recurrence relation

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + n$$

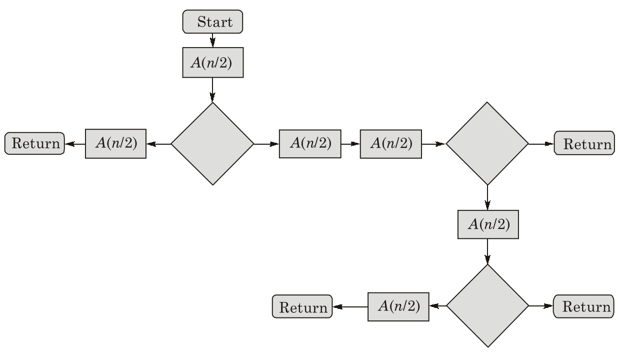
(a) $\theta(n)$ (b) $\theta(n^2)$
(c) $\theta(n \log n)$ (d) $\theta(n^3)$

Q.28 The solution of the Recurrence relation $T(m) = T(3m/4) + 1$ is

(a) $\theta(\log m)$ (b) $\theta(m \log m)$
(c) $\theta(m)$ (d) $\theta(\log \log m)$

Q.29 The given diagram shows the flow chart for a recursive function $A(n)$. Assume that all statements, except for the recursive calls, have $O(1)$ time complexity. If the worst case time complexity of this function is $O(n^{\alpha})$, then the least possible value (accurate up to two decimal positions) of α is ________.

Flow chart for Recursive Function $A(n)$



Q.30 What is complexity of recurrence relation

$$T(n) = T\left(\frac{n}{4}\right) + 7\left(\frac{3n}{4}\right) + n$$

(a) $\theta(n\log_3 n)$
(b) $\theta(n\log\log_{4/3} n)$
(c) $\theta(n^2 \log_{4/3} n)$
(d) $\theta(n\log_{4/3} n)$

Q.31 Which of the following can be solved using Master theorem?

(a) $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$

(b) $T(n) = 2T\left(\frac{n}{2}\right) + \log n$

(c) $T(n) = T\left(\frac{n}{2}\right) + \log n$

(d) None of these

Q.32 Consider the following recurrence relation:

$$T(n) = 7T\left(\frac{n}{2}\right) + an^2$$

What will be the solution of above recurrence?
(a) $0(n^2)$
(b) $0(n^3)$
(c) $0(n^2 \log n)$
(d) $0(n^{2.81})$

Q.33 What is complexity of recurrence relation

$T(n) = T(n-1) + 2n$ where $T(n) = 1$

(a) $\theta(n)$
(b) $\theta(n^2)$
(c) $\theta(n^3)$
(d) $\theta(n^2 \log n)$

Q.34 The running time of an algorithm $T(n)$, where n is the input size, is given by following:

$$T(n) = \begin{cases} 8T(n/2) + qn \text{ if } n > 1 \\ p \quad\quad\quad \text{if } n = 1 \end{cases}$$

where p and q are constants, the order of algorithm is
(a) n^2
(b) n^3
(c) n
(d) n^n

Q.35 We solve TOH problem recursively breaking the task in three sections, which of the following recurrence will accord will with the approach, that is shows correct order of work done on each recursive step.
(a) $T(n) = T(n-1) + 1 + T(n-1)$
(b) $T(n) = T(n-1) + 1T(n-1) + 1$
(c) $T(n) = 1 + T(n-1) + T(n-1)$
(d) $T(n) = T(n-1) + T(n-1) + 2$

Q.36 What is complexity of recurrence relation

$$T(n) = \sqrt{2}T\left(\frac{n}{2}\right) + C, \text{ for } n > 1$$
$$= a \text{ for } n = 1$$

(a) $\theta(\sqrt{n})$
(b) $\theta(\log n)$
(c) $\theta(n)$
(d) $\theta(\sqrt{n}\log n)$

Q.37 Let $T(n)$ be defined by $T(0) = T(1) = 4$ and $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + Cn$ for all integers $n >= 2$, where C is a positive constant. What is asymptotic growth of $T(n)$?
(a) $\theta(n)$
(b) $\theta(n \log n)$
(c) $\theta(n^2)$
(d) $\theta(n^{\log_3 4})$

Q.38 The recurrence equation:

$T(1) = 1$
$T(n) = 2T(n-1) + n, n \geq 2$

Evaluates to
(a) $2^{n+1} - n - 2$
(b) $2^n - n$
(c) $2^{n+1} - 2n - 2$
(d) $2^n + n$

Q.39 What will be time complexity for the following recurrence relation

$$T(n) = 8\sqrt{n}T(\sqrt{n}) + (\log n)^2$$

(a) $O(n \log n)$
(b) $O(n(\log n)^2)$
(c) $O(n(\log n)^3)$
(d) $O(n(\log n)^4)$

Q.40 Suppose $T(n) = 2T\left(\frac{n}{2}\right) + n$, $T(0) = T(1) = 1$

Which one of the following is FALSE?

(a) $T(n) = O(n^2)$ (b) $T(n) = \Theta(n \log n)$
(c) $T(n) = \Omega(n^2)$ (d) $T(n) = O(n \log n)$

Q.41 Solve the recurrence relation and find complexity of it

$$T(n) = T\left(\frac{n}{2}\right) + \frac{n^2}{2} + n$$

(a) $O(n^3)$ (b) $O(n)$
(c) $O(n^2)$ (d) $O(n^2 \log n)$

Q.42 An algorithm is made up of 2 modules $m1$ and $m2$. If order of $M1$ is $f(n)$ and $M2$ is $g(n)$ then the order of the algorithm is

(a) max $(f(n), g(n))$
(b) min $(f(n), g(n))$
(c) $f(n) + g(n)$
(d) $f(n) \times g(n)$

Q.43 Counting combinations can be also solved using divide and conquer approach

$$\left[\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}\right]$$

Consider the following function to compute n things chosen r at a time.

```
function X(n, r)
{   if (r = 0 or n = r) then return 1;
    else
    return (X(n – 1, r – 1) + X(n – 1, r));
}
```

Find the worst case time complexity of $X(n, r)$?

(a) $O(n)$ (b) $O(n^2)$
(c) $O(2^n)$ (d) None of these

Q.44 The recurrence relation:

$T(1) = 2$

$T(n) = 3T\left(\frac{n}{4}\right) + n$, has the solution then $T(n)$ equals to

(a) $O(n)$
(b) $O(\log n)$
(c) $O(n^{3/4})$
(d) None of these

Q.45 The time complexity of the following C function is (assume $n > 0$)

```
int recursive (int n )
{   if (n == 1)
       return (1);
       else
       return (recursive (n – 1) + recursive (n – 1));
}
```

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

Q.46 What is the time complexity of the following recursive function:

```
int DoSomething (int n)
{
      if (n < = 2)
            return 1;
      else
            return DoSomething (floor (sqrt (n)))+n;
}
```

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

Q.47 Solve the recurrence:

$$T(n) = T\left(\frac{n}{2} - 1\right) + T\left(n - \frac{n}{2}\right) + \theta(n)$$

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

Q.48 When $n = 2^{2k}$ for some $k \geq 0$, the recurrence relation

$T(n) = \sqrt{2}\,T\left(\frac{n}{2}\right) + \sqrt{n}$, $T(1) = 1$ evaluates to

(a) $\sqrt{n}(\log n + 1)$ (b) $\sqrt{n} \log n$
(c) $\sqrt{n} \log \sqrt{n}$ (d) $n \log \sqrt{n}$

Q.49 What is complexity of recurrence relation

$$T(n) = 3T\left(\frac{n}{2} + 47\right) + 2n^2 + 10n - \frac{1}{2}$$

(a) $O(n^2)$ (b) $O(n^{3/2})$
(c) $O(n \log n)$ (d) None of these

Q.50 Consider the following recurrence relation:

$$T(n) = 7T\left(\frac{n}{2}\right) + an^2$$

What will be the solution of above recurrence?

(a) $O(n^2)$ (b) $O(n^3)$
(c) $O(n^2 \log n)$ (d) $O(n^{2.81})$

Q.51 In Tower of Hanoi there are three towers: left, right and middle. There are n discs in left tower in a particular sequence (ascending order), we have to transfer these discs into right tower having same sequence using middle tower (consider that disc with smaller sequence number is always at top of large sequence number i.e. 1 is above 2 but 2 above 1 is not allowed). If Towers of Hanoi is implemented with recursive function then total discs moves to move 9 discs from left to right are _______.

Q.52 Let m, n be positive integers. Define $Q(m, n)$ as

$Q(m, n) = 0$, if $m < n$

$Q(m - n, n) + p$, if $m \geq n$

Then $Q(m, 3)$ is (a div b, gives the quotient when a is divided by b)

(a) a constant
(b) $p \times (m \bmod 3)$
(c) $p \times (m \text{ div } 3)$
(d) $3 \times p$

Q.53 Find complexity of recurrence relation

$T(n) = T(n-1) + T(n-2) + C$

(a) $O(2^n)$
(b) $O(n^2)$
(c) $O(n)$
(d) $O(n^n)$

Q.54 The recurrence relation capturing the optimal execution time of the Towers of Hanoi problem with n discs is

(a) $T(n) = 2T(n-2) + 2$
(b) $T(n) = 2T(n-1) + n$
(c) $T(n) = 2T\left(\frac{n}{2}\right) + 1$
(d) $T(n) = 2T(n-1) + 1$

■■■■

Answers Recurrence Relations

1.	(c)	2.	(c)	3.	(d)	4.	(c)	5.	(a)	6.	(a)	7.	(a)	8.	(a)	9.	(a)
10.	(a)	11.	(c)	12.	(b)	13.	(c)	14.	(c)	15.	(b)	16.	(b)	17.	(a)	18.	(b)
19.	(c)	20.	(d)	21.	(b)	22.	(a)	23.	(a)	24.	(b)	25.	(a)	26.	(a)	27.	(a)
28.	(a)	29.	(2.32)	30.	(d)	31.	(b)	32.	(d)	33.	(b)	34.	(b)	35.	(a)	36.	(a)
37.	(a)	38.	(a)	39.	(c)	40.	(c)	41.	(c)	42.	(a)	43.	(c)	44.	(a)	45.	(d)
46.	(d)	47.	(a)	48.	(a)	49.	(a)	50.	(d)	51.	(511)	52.	(c)	53.	(a)	54.	(d)

Explanations Recurrence Relations

1. (c)

$[n(\log(n^3) - \log n) + \log n]\, n + \log n$

$$= \left[n\left(\log\frac{n^3}{n}\right) + \log n\right] n + \log n$$

$= [n \log n^2 + \log n] n + \log n$

$= n^2 . 2 \log n + n \log n + \log n$

$= 2n^2 \log n + n \log n + \log n = O(n^2 \log n)$

So option (c) is correct.

2. (c)

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

By Repetitive Substitution

$T(n) = [T(n-2) + n - 1] + n$

$= [(n-2) + T(n-3] + (n-1) + n$

$= 0 + 1 + 2 + \ldots + n$

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

So option (c) is correct.

3. (d)

Since, we have a square root term, considering only perfect squares and those which are multiple of 2 as that can take care of log.

$T(2) = 1$ //Assume

$T(2^2) = T(2) + 1 = 2$

$T\left(2^{2^2}\right) = T(4) + 1 = 3$

$T\left(2^{2^3}\right) = T(16) + 1 = 4$

$T\left(2^{2^4}\right) = T(256) + 1 = 5$

So, we are getting

$$T(n) = \log\log n + 1$$

$$\Rightarrow \quad T(n) = O(\log\log n)$$

4. (c)

From Master theorem, case 2

Here, $a = \sqrt{2}$, $b = 2$, $k = 1/2$, $p = 0$

$a = b \wedge k$ which is true.

$p > -1$

So, complexity is theta ($n \wedge$ (log a base to b) log $\wedge$ $(p + 1)n$

$$= \theta(\sqrt{n}(\log n + 1))$$

5. (a)

$$T(n) = T\left(\frac{2n}{3}\right) + 1$$

By using recursive tree method:

$$T(n)$$
$$\downarrow$$
$$T\left(\frac{2n}{3}\right) = O(1)$$
$$\downarrow$$
$$T\left(\frac{2^2 n}{3^2}\right) = O(1)$$
$$\downarrow$$
$$T\left(\frac{2^3 n}{3^3}\right) = O(1)$$
$$\vdots$$
$$T\left(\frac{2^k n}{3^k}\right) = O(1)$$

So, height of tree is $\log_{(3/2)} n$.

So, $T(n)$ = Height × Work at each level

$$= \log_{3/2} n \times O(1) = O(\log_{3/2} n)$$

6. (a)

$$T(n) = \begin{cases} T(n-1) + T(n-2) - T(n-3) & n > 3 \\ n & \text{otherwise} \end{cases}$$

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

$= 3 + 2 - 1 = 4$

$T(5) = T(4) + T(3) - T(2)$

$= 4 + 3 - 2 = 5$

So by induction,

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

$= n - 1 + n - 2 - (n - 3)$

$= 2n - 3 - n + 3 = n$

So, $T(n) = O(n)$

7. (a)

Let, $T(1) = T(2)$

$= T(3) = k$ (say)

Then, $T(4) = k + k - k = k$

$T(5) = k + k - k = k$

By mathematical induction it can be proved that $T(n) = k$, a constant.

8. (a)

In these type of questions, where master theorem does not apply you can do the following

$$T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{3}$$

Reframe it like this, by putting i in the specified positions.

$$T(n) = 2^i T\left(\frac{n}{4^i}\right) + i\sqrt{3}$$

Now, for $\frac{n}{4^i} = 1$, put $i = \log_4 n$

So, the equation becomes

$$T(n) = 2^{\log_4 n} + \log_4 n\sqrt{3}$$

or

$$T(n) = n^{\log_4 2} + \log_4 n\sqrt{3}$$

$$T(n) = \sqrt{n} + \log_4 n\sqrt{3}$$

and therefore, $T(n) = O(\sqrt{n})$

9. (a)

Take base conditions as $T(0) = 1$ and $T(-1) = 0$.

$T(-1) = 0$

$T(1) = 2T(0) - T(-1) = 2$

$T(2) = 2T(1) - T(0)$

$= 3$ ... and so on

So, option (a) is correct.

10. (a)

$$T(n) = 2.T\left(\frac{n}{2}\right) + \log n$$

[By Master's Theorem case1]

$$= \Theta(n)$$

11. (c)

The time complexity of computing the transitive closure of a binary relation on a set of n-element will take $O(n^3)$ using Floyd-Warshall's algorithm.