POSTAL Book Package 2021

Computer Science & IT

Objective Practice Sets

Algorithms

Contents

MADE EASY Publications

2 CHAPTER
Recurrence Relations

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:

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

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

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

(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
$T(n) = \text{sqrt}(2) + (n/2) + \text{sqrt}(n)$
(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 $T(n) = T(2n/3) + 1$ 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 $n \geq 1$.
(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.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 recurrance relation

$$T(n) = 7T(n/2) + an^2$$

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.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 no. is always at top of large sequence no. 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(n/2) + 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 sqrt term, considering only perfect squares and those which are multiple of 2 as that can take care of log.

$$T(2) = 1 \quad \text{//Assume}$$
$$T(2^2) = T(2) + 1 = 2$$
$$T\left(2^{2^2}\right) = T(4) + 1 = 3$$
$$T\left(2^{2^3}\right) = T(16) + 1 = 4$$
$$T\left(2^{2^4}\right) = T(256) + 1 = 5$$

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 ^ (log a base to b) log ^ $(p + 1)n$

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

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

$$= T(3) = k \text{ (say)}$$

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.

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

Now, for $n/4^i = 1$, 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 $T(-1) = 0$.

$$T(-1) = 0$$
$$T(1) = 2T(0) - T(-1) = 2$$
$$T(2) = 2T(1) - T(0)$$
$$= 3 \text{ ... and so on}$$

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.

12. (b)

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

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

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

$$T(n-3) = T(n-4) + \frac{1}{n-3}$$

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

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

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

$$T(n) = T(n-k) + \frac{1}{(n-(k-1))} + \frac{1}{(n-(k-1))} + \frac{1}{(n-(k-3))} + \frac{1}{(n-(k-4))} + \ldots + \frac{1}{n}$$

$$T(n) = T(n-k) + \frac{1}{(n-k+1)} + \frac{1}{(n-k+2)} + \frac{1}{(n-k+3)} + \ldots + \frac{1}{n-3} + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}$$

Thing that the one condition is given if $(n > 1)$

So, we can substitute $(k = n - 1)$

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

$$T(n) = T(1) + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \ldots + \frac{1}{n}$$

Assume that $T(1) = 1$.

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

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

13. (c)

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

Put $n = 2^m$, $\because \quad m = \log n$

$$T(2^m) = T(\sqrt{2^m}) + 1$$

$$\underset{S}{\boxed{T(2^m)}} = \underset{S}{\boxed{T(2^{m/2})}} + 1$$

$$S(m) = S(m/2) + 1$$

Using Master's Theorem

$$S(m) = \log m \quad (\because \quad m = \log n)$$
$$T(n) = O(\log \log n)$$

14. (c)

Using Master Theorem

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$$n^{\log_b^a} = n^{\log_2^8} = n^3$$
$$f(n) = n^2 \log n$$

If $n^{\log_b^a} > f(n)$ then

$$T(n) = O(n^{\log_b^a})$$

$\Rightarrow \quad n^3 > n^2$

$\Rightarrow \quad T(n) = O(n^3)$

So option (c) is correct.

15. (b)

According to Master theorem

$a = 7$, $b = 3$, $k = 2$

So, its case of $a < b^k$

So, $O(N^2)$ will be complexity

So, option (b) is correct.