POSTAL Study Course 2018

Computer Science & IT

Objective Practice Sets

Algorithm

Contents

MADE EASY

India's Best Institute for IES, GATE & PSUs

For MADE EASY Students Only

1

CHAPTER

Computer Science & IT

Asymptotic Analysis of Algorithms

Q.1 Which of the following standard algorithms is not a greedy algorithm?
(a) Selection sort
(b) Kruskal's algorithm
(c) Bellman ford shortest path algorithm
(d) Dijkstra's shortest path algorithm

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 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.4 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

Q.5 Consider the following functions.

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

Identify the functions in increasing order of growth.
(a) $\frac{n}{\log n}$, $\log(\log n)$, $(\log n)$, $(\log n)^2$, $\sqrt{n}$, $\sqrt{n}\cdot\log n$, n
(b) $\log(\log n)$, $\log n$, $(\log n)^2$, $\sqrt{n}$, $\sqrt{n}\log n$, $\frac{n}{\log n}$, n, $n\log n$
(c) $\log(\log n)$, $\log n$, $(\log n)^2$, $\sqrt{n}$, $\frac{n}{\log n}$, $\sqrt{n}\log n$, n, $n\log n$.
(d) None of these

Q.6 Compute the running time of the following *C* code. Assume that $n \geq 1$ and time complexity as a function of n.

```
x = n;
while (x > 0)
{   y = n;
    while (y > 0)
    {   sum = sum + 10;
        y = y/2;
    }
    x = x - 10;
}
```

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

Q.7 Consider the following code

```
Sum = 0;
for (i = 1 to n)
  for (j = i to 2n)
    Sum = Sum + 1;
```

Find the time complexity of above code in terms of n.

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

Q.8 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.9 Determine the asymptotic running time of $f(n)$ for the following C program.

```
int f(int n)
{
    int x;
    if (n == 0) x = 1;
    else x = f(n – 1)*2;
    g(x);
    return x;
}
void g(int m)
{
    int y;
    for (y = m; y > 0; y/ = 2);
}
```

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

Q.10 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.11 Consider the following C function

```
find (int n)
{   if (n < 2) then return;
    else
    {  sum = 0;
       for (i = 1; i ≤ 4; i++)
      find (n/2);
      for (i = 1; i ≤ n*n; i ++)
       sum = sum +1;
    }
}
```

Assume that the division operation takes constant time and "sum" is global variable. What is the time complexity of "find (n)" ?

(a) $\Theta(n^2)$
(b) $\Theta(n^2 \log n)$
(c) $\Theta(n^3)$
(d) None of these

Q.12 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.13 Consider the following two functions.

$$f(n) = n^3, \text{ if } 0 \le n < 10{,}000$$
$$\quad\; n^2, \text{ otherwise}$$
$$g(n) = n, \text{ if } 0 \le n < 100$$
$$\quad\; n^2 + 5n, \text{ otherwise}$$

Which of the following is/are true?

1. $f(n)$ is $O(n^3)$
2. $g(n)$ is $O(n^3)$
3. $O(f(n))$ is same as $O(g(n))$
4. $g(n)$ is $O(n^2)$

(a) 3 and 4
(b) 1, 2 and 3
(c) 1, 2 and 4
(d) None of these

Q.14 Look at the following algorithm

```
int n;
int A [100];
void x ( )
{   int i;
    for (i = n/2; i > = 1; i – –)
    y(i);
    while (n > 1) {
    y(1); }
}
void y(int i)
{   ..........;
    ..........;
    ..........;
}
```

Let complexity of y is $O(\log_2 n)$. Then the complexity of x will be

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

Q.15 Let $f(n) = \Omega(n)$, $g(n) = O(n)$ and $h(n) = \Theta(n)$. Then $[f(n) \cdot g(n)] + h(n)$ is ________.

(a) $\Omega(n)$
(b) $O(n)$
(c) $\theta(n)$
(d) None of these

Answers Asymptotic Analysis of Algorithms

1.	(c)	2.	(c)	3.	(c)	4.	(b)	5.	(b)	6.	(b)	7.	(c)	8.	(b)	9.	(b)
10.	(d)	11.	(b)	12.	(c)	13.	(a)	14.	(c)	15.	(a)						

Explanations Asymptotic Analysis of Algorithms

1. (c)

Bellman ford shortest path algorithm is example of dynamic programming.

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. (c)

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

4. (b)

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{1}{k}\right) = \frac{1}{k}\sum_{i=1}^{n} \sum_{j=i+1}^{n} (1)$$

$$= \frac{1}{k}\sum_{i=1}^{n}\left[1+1+1+\ldots+n-(i+1)+1 \text{ times}\right]$$

$$= \frac{1}{k}\sum_{i=1}^{n}[n-i]$$

$$= \frac{1}{k}\left[n\sum_{i=1}^{n}(1) - \sum_{i=1}^{n}(i)\right] = \frac{1}{k}\left[n \cdot n - \frac{n(n+1)}{2}\right]$$

$$= \frac{1}{k}\left[n^2 - \frac{n^2+n}{2}\right] = \frac{1}{2k}\left[n^2 - n\right] = O(n^2)$$

5. (b)

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

So option (b) is correct.

6. (b)

Inner while loop executes $\log_2^n$ times for every x.

$$y = n, \frac{n}{2}, \frac{n}{4}, \ldots \frac{n}{n} \Rightarrow \log_2 n \text{ times}$$

outer loop: $x = n, n-10, n-20, \ldots 1$

$$\Rightarrow \quad O\left(\frac{n}{10}\right) = O(n)$$

$\therefore$ Time complexity = $O(n\log_2 n)$

So option (b) is correct.

7. (c)

Assume that each simple statement executes in constant time.

$$= \sum_{i=1}^{n}\sum_{j=i}^{2n}(1) = \sum_{i=1}^{n}(1+1+1+\ldots+(2n-i+1)\text{times})$$

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

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

$$= \frac{(4n^2 - n^2 - n + 2n)}{2} = \frac{(3n^2+n)}{2} = O(n^2)$$

So option (c) is correct.

8. (b)

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

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

9. (b)

Time complexity of $g(m)$ is : $\Theta(\log m)$

[***Note:*** f(n) returns the value 2^n]

$$g(x) = g(2^n) = g(m) \Rightarrow m = 2^n$$

$$\therefore \quad g(m) = \Theta(\log m) = \Theta(\log 2^n) = \Theta(n)$$

$$\therefore \quad g(m) = \Theta(n)$$

Recurrence relation for $f(n)$:

$$f(n) = f(n-1) + g(m)$$

$$= f(n-1) + \Theta(n) = \Theta(n^2)$$

10. (d)

$n^k < a^n < e^n < n! < n^n$

n^n will take maximum asymptotic time.

$n! = 0(n^n)$

$e < n \qquad \therefore e^n < n^n$

$k < n \qquad \therefore n^k < a^n$

11. (b)

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

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

Using master's theorem

$$n^{\log_b a} = n^{\log_2 4} = 2$$

$$n^2 = n^2$$

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

Case 2 is applied. So, option (b) is correct.

12. (c)

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

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

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

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

13. (a)

In this question total possibility is four but only two possibility will give solution and other two possibilites will not give solutions.

$g(n) = O(f(n))$ in case of $0 \le n < 100$ & & $0 \le n < 10{,}000$ (it is also not possible)

$g(n) = \Theta(f(n))$ otherwise

From the above two equations the conclusion is f(n) is same as $O(g(m))$ and $g(n)$ and $f(n)$ will be $O(n^2)$

14. (c)

In this program, for loop run $n/2$ times and in this 'for loop' $y(i)$ have ($\log_2 n$) time complexity.

In the above we will select maximum time complexity i.e. $\frac{n}{2}\log_2 n$

Then the total time complexity for the 'for loop' will O(nlogn)

15. (a)

$$f(n) = \Omega(n) \Rightarrow f(n) \ge c \cdot n$$

$$g(n) = O(n) \Rightarrow g(n) \le c \cdot n$$

$$h(n) = \Theta(n) \Rightarrow c_1 \cdot n \le h(n) \le c_2 \cdot n$$

$$f(n) \cdot g(n) = c \cdot n$$

$$[\because \quad f(n) \ge c \cdot n \text{ and } g(n) \le c \cdot n]$$

$$\underbrace{f(n).g(n)}_{\ge c.n} + \underbrace{h(n)}_{\theta(n)} = \Omega(n)$$

2 CHAPTER

Recurrence Relations

Q.1 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.2 Consider the following recurrance relation

$T(n) = 8T\left(\frac{n}{2}\right) + n^2 \log n$, $n > 1$ 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.3 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.4 Match **List-I** (Divide Conquer) with **List-II** (Recurrence Relation (Worst-Case) and select the correct answer using the codes given below the lists:

List-I	List-II
A. Binary search	**1.** $T(n) = 2T(n/2) + C$
B. Merge sort	**2.** $T(n) = T(n/2) + 1$
C. Quick sort	**3.** $T(n) = T(n-1) + n - 1$
D. Max Min	**4.** $T(n) = 2T(n/2) + kn$

Codes:

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

Q.5 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.6 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.7 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.8 Let $T(n) = 2T\left(\frac{n}{4}\right) + 100\sqrt{n}$. Find time complexity of $T(n)$?

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

Q.9 Let $T(n) = T(n - 1) + 1/n$. Then $T(n)$ is

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

Q.10 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.11 Consider the following recurrance relation

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

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.12 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.13 Which of the following represent correct recurrence for worst case of Binary search with n element.

(a) $T(n) = T(n/2) + O(n)$

(b) $T(n) = 2T(n/2) + O(1)$

(c) $T(n) = 2T(n/2) + O(n)$

(d) $T(n) = T(n/2) + O(1)$

Q.14 The recurrence relation

$T(1) = 2$

$T(n) = 3T(n/4) + n$, 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.15 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. ie. 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 _______.

■■■

Answers Recurrence Relations

1.	(c)	2.	(c)	3.	(c)	4.	(d)	5.	(a)	6.	(c)	7.	(c)	8.	(b)	9.	(b)
10.	(b)	11.	(d)	12.	(b)	13.	(d)	14.	(a)								

Explanations Recurrence Relations

1. (c)

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

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

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

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

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

Using Master's Theorem

$$S(m) = \log m \qquad (\because m = \log n)$$

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

2. (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.

3. (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.

4. (d)

Binary Search: $T(n) = T(n/2) + 1$ [Avg, Worst]

Merge Sort: $T(n) = 2T(n/2) + kn$ [Avg, Best, Worst]

Quick Sort: $T(n) = T(n-1) + n - 1$ [Worst Case]

Max Min: $T(n) = 2T(n/2) + C$ [Best, Avg, Worst]

5. (a)

$$T(n) = T(n/5) + T(7n/10) + n$$

$$= \sum_{i=0}^{\log_5^n}\left(\frac{9}{10}\right)^i \cdot n = \Theta(n)$$

6. (c)

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

By Repetitive Substitution

$$\begin{aligned} T(n) &= [T(n-2) + n - 1] + n \\ &= [(n-2) + T(n-3] + (n-1) + n \\ &= 0 + 1 + 2 + \ldots + n \\ &= \frac{n(n+1)}{2} = \frac{n^2+n}{2} \end{aligned}$$

So option (c) is correct.

7. (c)

$$\begin{aligned} T(n) &= T(n-1) + T(n-1) + d \\ &= 2T(n-1) + d \\ &= 2^i T(n-i) + d\sum_{j=0}^{i-1} 2^j \end{aligned}$$

[Solving by Substitution $\because i = n - 1$]

$$= 2^{n-1} T(1) + d\sum_{j=0}^{n-2} 2^j$$

$$\left[\because \sum_{r=0}^{n-1} a^r = \frac{a^n - 1}{a-1}\right]$$

$$\begin{aligned} &= 2^{n-1} \cdot c + d\frac{(2^{n-1}-1)}{2-1} \\ &= 2^{n-1}\ .\ c + d.2^{n-1} - d \\ &= (c+d)2^{n-1} - d = O(2^n) \end{aligned}$$

8. (b)

$$\begin{aligned} n^{\log_b^a} &= n^{\log_4^2} = n^{\frac{1}{2}} = \sqrt{n} \\ f(n) &= 100\sqrt{n} \\ n^{\log_b^a} &= \Theta(f(n)) \\ \therefore\quad T(n) &= \Theta\left(\sqrt{n}\log n\right) \end{aligned}$$

So option (b) is correct.

9. (b)

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

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

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

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

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

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

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

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

10. (b)

$$T(n) = 2.T(n/2) + \log n$$

$$\because\quad \log n = O(n^{\log_b a-\epsilon}) \text{ for some } \varepsilon > 0$$

$$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$$

$$\therefore\quad T(n) = \Theta(n)$$

So option (b) can be solved using master theorem.

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

Try case 1: $\dfrac{n}{\log n} \neq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$,

because $\dfrac{1}{\log n} \neq O(n^{-\varepsilon})$

Try case 2: $\dfrac{n}{\log n} \neq \Theta(n^1)$, because $\dfrac{n}{\log n} = o(n)$

Try case 3: $\dfrac{n}{\log n} \neq \Omega(n^{1+\varepsilon})$ for some $\varepsilon > 0$,

because $\dfrac{n}{\log n} = o(n)$

All cases failed using master theorem for option (a), similarly option (c) is also not applicable.

11. (d)

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

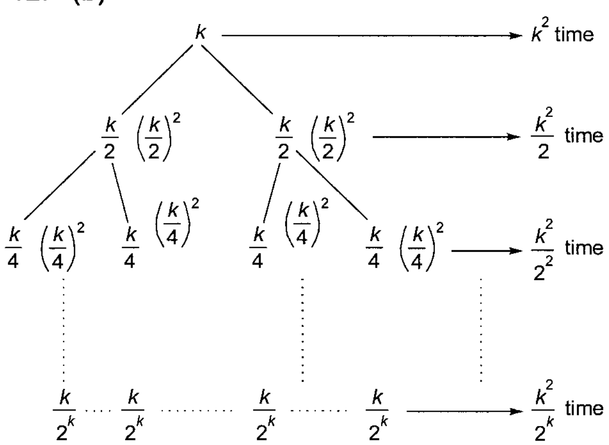
using Master's method
$a = 7$, $b = 2$, $f(n) = an^2$

$$h(n) = \frac{f(n)}{n^{\log_b a}} = \frac{an^2}{n^{\log_2 7}} = \frac{an^2}{n^{2.81}}$$
$$= an^{-0.81}$$

So $\quad U(n) = 0(1)$

$$T(n) = n^{\log_b a}U(n) = n^{\log_2 7}0(1)$$
$$T(n) = 0(n^{2.81})$$

12. (b)



$$T(n) = \frac{k^2}{2^0} + \frac{k^2}{2^1} + \frac{k^2}{2^2} + \ldots \frac{k^2}{2^k}$$

$$T(k) = k^2\left[1 + \frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^k}\right]$$
$$= k^2 \times O(1) = O(k^2)$$

13. (d)

In Binary search we first compare the given element with middle element of the array. If the element matches with middle element, then return middle element index. Otherwise, go to left half of array or right half of array.

14. (a)

Let $\quad 4^k = n$

$$T(n) = 3T(n/4) + n$$

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

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

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

$$= 3^kT(n/4^k) + n + \frac{n}{4} + \frac{n}{4^2} + \ldots \frac{n}{4^{k-1}}$$

$$= 3^kT(1) + n\left[1 + \frac{1}{4} + \frac{1}{4^2} + \frac{1}{4^3} + \ldots\right]$$

$$= 3^k \cdot 2 + n \times \left(\frac{\left(\frac{1}{4}\right)^k - 1}{\frac{1}{4} - 1}\right) \cong O(n)$$

15. (511)

The recurrence relation for tower of Hanoi problem is:

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

$\Rightarrow$ Number of moves

$$2^n - 1 = 2^9 - 1 = 511$$

■■■■

Divide and Conquer

Q.1 Suppose there are 4 sorted lists of $\frac{n}{2}$ elements each. If we merge these lists into a single sorted list of n elements, how many key comparisons are needed in the worst case using an efficient algorithm?

(a) $\frac{6}{4}n-3$ (b) $\frac{7}{4}n-3$
(c) $\frac{8}{4}n-3$ (d) $\frac{9}{4}n-3$

Q.2 The best case of quick sort helps Vivek to sort a particular data set of size 'n' is 2048 msec. Gaurav also tried the same algorithm on similar data set and it took him 324 msec in worst case to sort a file of size 18. The file size of Vivek is________.

Q.3 Consider the following sorting algorithms
A. Merge sort **B.** Quick sort
C. Heap sort **D.** Insertion sort
Find the correct pair such that their best, average and worst case time complexities are same. (Assume that all elements are distinct)
(a) A, B (b) A, C
(c) A, D (d) None of these

Q.4 What is time complexity of matrix multiplication (Strassen's) using divide and conquer approach?
(a) $O(n^{\log 5})$ (b) $O(n^{\log 6})$
(c) $O(n^{\log 7})$ (d) None of these

Q.5 A list has n-strings where each string is of length n. Using divide and conquer approach, what is the worst case running time (tightest upper bound) to find the first string?
Note: First string is a string which appears first when all strings are sorted into the lexicographical order.
(a) $O(n^2)$ (b) $O(n)$
(c) $O(n\log n)$ (d) $O(n^2 \log n)$

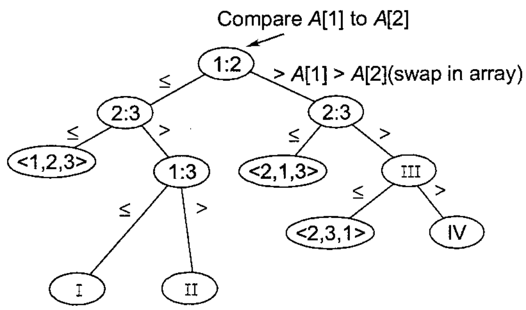
Q.6 Consider the following sorting algorithm.

```
Sorting (A, low, high)
{
    if (low = high) return;
    if (low+1 = high)
    {
        if (A[low] > A[high])
        Swap (A[low], A[high]);
        return;
    }
    t1 = low + ((high - low + 1)/3);
    t2 = low + 2·((high - low + 1)/3);
    Sorting (A, low, t2);
    Sorting (A, t1, high);
    Sorting (A, low, t2);
}
```

What is the running time of Sorting(A, 1, n) function.
(a) $\Theta(n^{1.7})$ (b) $\Theta(n^{2.7})$
(c) $\Theta(n^{3.7})$ (d) $\Theta(n^{0.7})$

Q.7 Which of the following is/are true?
S1 : A sorting algorithm is in place if the relative order of common element is maintained after sorting.
S2 : A sorting algorithm is stable if it requires very little additional space besides the initial array holding the elements that are to be sorted.
(a) Only S1 is true
(b) Only S2 is true
(c) Both S1 and S2 are true
(d) Both S1 and S2 are false

Q.8 For insertion sort on 3 elements consider the following decision tree



Each internal node is labelled by indices of array elements from their original positions. Each leaf is labelled by the permutations of orders that the algorithm determines.

Fill in the blanks for I, II, III, IV

(a) <3, 1, 2>, <1, 3, 2>, (3 : 2), <3, 2, 1>
(b) <1, 3, 2>, <3, 1, 2>, (2 : 3), <3, 2, 1>
(c) <3, 1, 2>, <3, 2, 1>, (3 : 2), <1, 3, 2>
(d) <1, 3, 2>, <3, 2, 1>, (2 : 3), <3, 1, 2>

Q.9 Give the result of partitioning the keys.

T H I S C O U R S E I S O V E R

after the 1st pass of quick sort. Choose the last elements as pivot element (R). Also for duplicates, adopt the convention that both pointers stop.

(a) E H I O C O I E R R U S S V T S
(b) E H I S C O I E R R U S O V T S
(c) E H I O C O U E S R T S S V T R
(d) E H I O O C I E R R U S S V T S

Q.10 Which of the following cases is definite to give O(n log n) time complexity always for quick sort to sort the array A[0...n], for any set of values of array A.

Case 1: Choosing middle element as pivot.
Case 2: Choosing pivot element as $(2^0)^{th}$ element initially followed by $(2^1)^{th}$ element, followed by $(2^2)^{th}$ element of array A and so on.
Case 3: Choosing median element as pivot.
Case 4: Choosing the pivot element randomly from the array A.
Case 5: Choosing pivot such that the array is partioned into almost two equal subarrays.

(a) Only 1 and 4
(b) Only 1, 2 and 3
(c) Only 3 and 5
(d) Only 1, 3 and 4

Q.11 Consider an array consisting of the following elements in unsorted order (placed randomly), but 60 as first element.

60, 80, 15, 95, 7, 12, 35, 90, 55

Quick sort partition algorithm is applied by choosing first element as pivot element. How many total number of arrangements of array integers is possible preserving the effect of first pass of partition algorithm.

Q.12 Find the number of comparisons that will be needed in worst case to merge the following sorted files into a single sorted file by merging together two files at a time?

Files	$f1$	$f2$	$f3$	$f4$
Number of records	20	21	22	23

Q.13 Merging 4 sorted files having 200, 50, 125, 25. records will take 0 (–) time?

Q.14 You want to check whether a given set of items is sorted. Which of the following sorting methods will be the most efficient if it is already in sorted order?

(a) Bubble sort
(b) Selection sort
(c) Insertion sort
(d) Merge sort

Q.15 A machine needs a minimum of 100 sec to sort 1000 names by quick sort. The minimum time needed to sort 100 names will be approximately

(a) 50.2 sec
(b) 6.7 sec
(c) 72.7 sec
(d) 11.2 sec

Q.16 If one uses straight two-way merge sort algorithm to sort the following elements in ascending order

20, 47, 15, 8,9, 4, 40, 30, 12, 17

then the order of these elements after the 2nd pass of the algorithm is

(a) 8, 9, 15, 20, 47, 4, 12, 17, 13, 40
(b) 8, 15, 20, 47, 4, 9, 30, 40, 12, 17
(c) 15, 20, 47, 4, 8, 9, 12, 30, 40, 17
(d) 4, 8, 9, 15, 20, 47, 12, 17, 30, 40

Q.17 Which of the following is the average no. of key comparisons done by sequential search in the successful case?

(a) $(n+1)/2$ (b) $n/2$
(c) $(n+1)$ (d) $2n$

Q.18 Given an array A of elements:

80, 70, 55, 90, 71, 72, 73, 77, 52, 50

What will be the output after 1 pass of 3 way merge sort?

(a) 70, 80, 55, 90, 71, 72, 73, 77, 50, 52
(b) 55, 70, 80, 71, 72, 90, 52, 73, 50, 77
(c) 50, 52, 55, 70, 71, 72, 73, 77, 80, 90
(d) 55, 70, 80, 71, 72, 90, 52, 73, 77, 50

Q.19 Given an array of n elements, each of which is at most k positions from its target postion, if merge sort is used to sort such an an array what is the time complexity?

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

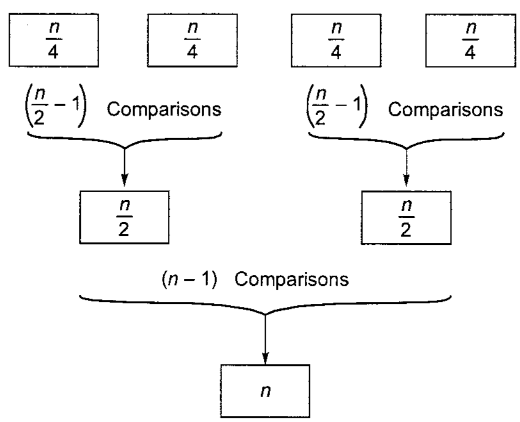
■■■

Answers Divide and Conquer

1.	(c)	3.	(b)	4.	(c)	5.	(c)	6.	(b)	7.	(d)	8.	(b)	9.	(a)	10.	(c)
14.	(c)	15.	(b)	16.	(b)	17.	(a)	18.	(d)	19.	(b)						

Explanations Divide and Conquer

1. (c)



$$\text{Total comparisons} = \left(\frac{n}{2}-1\right)+\left(\frac{n}{2}-1\right)+(n-1)$$

$$= 2n-3 = \frac{8}{4}n-3$$

2. (256)

Best case of quick sort is $O(n \log n)$ and it takes 2048 msec.

$$2048 = c_1 \cdot n \log n \qquad \ldots(1)$$

Worst case of quick sort is $O(n^2)$ and it takes 324 msec.

$$324 = c_1 \cdot n^2 \quad [n = 18]$$

$$324 = c_1 \cdot 18^2$$

$$c_1 = 1$$

Substitute c_1 n = 1 in equation (1)

$$c_1 \cdot n \log n = 2048 \quad [c_1 = 1]$$

$$n \log n = 2048$$

Put $n = 2^K$

$$2^K \times K = 2048 \qquad \ldots(2)$$

The value of K = 8 which satisfies equation (2)

$\because$ Vivek's file size = $2^K = 2^8 = 256$

3. (b)

Merge sort and Heap sort have $O(n \log n)$ time complexity.

4. (c)

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

Using Master Theorem $[n^{\log_2 7} > kn^2]$

$$T(n) = O(n^{\log_2 7})$$

5. (c)

To find the first string, algorithm is similar to MaxMin algorithm using divide and conquer.

$$T(n) = 2.T(n/2) + O(n)$$

{$O(n)$ is to find the first string from any two subproblems}

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

6. (b)

$$\begin{aligned} T(n) &= 0, \text{ if } n = 1 \\ &= 1, \text{ if } n = 2 \\ &= 3.T(2n/3) + 2;\ n \geq 3 \\ &= \Theta(n^{2.71}) \end{aligned}$$

[Using Master theorem]

7. (d)

Stable Sorting : A sorting algorithm is stable if the relative order of common element is maintained after sorting.

In-place sorting : A sorting algorithm is inplace if it requires very little additional space besides the initial array holding the elements that are to be sorted.

For **Ex.** Quick sort is not stable sorting but it is in place sorting.

8. (b)

x	y	z
A[1]	A[2]	A[3]

compare x to y i.e. $A[1]$ to $A[2]$

if $x \leq y$

then remain as it is and compare y and z i.e. $A[2]$ to $A[3]$

Else compare x to z i.e. $A[1]$ to $A[3]$

Similarly apply the insertion sorting operation.

9. (a)

T H I S C O U R S E I S O V E R ∞

pivot

T H I S C O U R S E I S O V E R ∞

i j

stop i as you found $a[i] \leq$ pivot

stop j as you found $a[j] <$ pivot

if $i < j$ then swap $a[i]$ and $a[j]$

if $i > j$ then swap $a[i]$ and pivot

E T

~~T~~ H I S C O U R S E I S O V ~~E~~ R ($i < j$)

i j

O S

E H I ~~S~~ C O U R S E I S ~~O~~ V T R ($i < j$)

i j

I U

E H I O C O ~~U~~ R S E ~~I~~ S S V T R ($i < j$)

i j

E R

E H I O C O I ~~R~~ S ~~E~~ U S S V T R ($i < j$)

i j

R S

E H I O C O I E ~~S~~ R U S S V T ~~R~~ ($i > j$)

j i

E H I O C O I E R R U S S V T S

10. (c)

Only case 3 and 5 ensures O(n log n) time complexity always.

Case 1: Choosing middle element doesn't mean that pivots location is in middle, after the partition algorithm. In worst case pivot may go either first or last location ($O(n^2)$). Similarly Case 2, Case 4 does not guarantee $O(n \log n)$.

But cases 3 and 5 are similar and makes sure that pivot always goes to middle location, after the partition algorithm.

11. (720)

We have to choose first element as pivot.

Here 60 is given as first element.

After first pass, the pivot element goes to it's exact location.

Here 60 goes to 6^{th} place.

All the elements less than 60 go to left of 60 and all the elements greater than 60 go to right of 60.

After 1^{st} pass

$$\underbrace{55 \quad 15 \quad 7 \quad 12 \quad 35}_{5!} \quad 60 \quad \underbrace{80 \quad 90 \quad 95}_{3!}$$

$\Rightarrow$ $5! \times 3!$

$\Rightarrow$ 720 possible arrangements.

12. (169)

Given files

$\underline{20, 21}$, 22, 23 } 41 – 1 = 40 comparisons

41, $\underline{22, 23}$ } 45 – 1 = 44 comparisons

$\left.\begin{array}{c} 41, 45 \\ \hline 85 \end{array}\right\}$ 86 – 1 = 85 comparisons

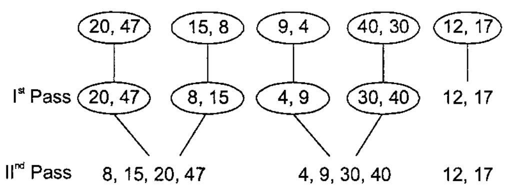
40 + 44 + 85 = 169

13. (400)
Two sorted file of size m and n take $O(m + n)$ time for merging so total time = 200 + 50 + 125 + 25 = 400

14. (c)
Insertion sort in best case = $O(n)$
and bubble sort = $O(n^2)$; selection sort = $O(n^2)$;
merge sort = $O(n \log n)$

15. (b)
In the best case quick sort algorithm makes $n \log (n)$ comparisons. So $1000 \times \log (1000) =$ 9000 comparisons, which takes 100 sec. To sort 100 names a minimum of 100 (log 100) = 600 comparisons are needed.
This takes $100 \times 600/9000 = 6.7$ sec.

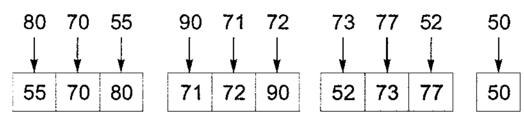
16. (b)



Hence (b) is the correct option.

17. (a)
In linear or sequential search the maximum no. of comparisons are (n) for n elements hence the no. of comparisons when the element to be found is at the middle of the array
$= (n + 1)/2$ [if n is odd]
$= n/2$ [if n is even]
in general average case = $(n + 1)/2$
Hence (a) is the correct option.

18. (d)
3 way merge sort will combine 3 subarrays at a time. After division elements becomes.



19. (b)
Divide the elements into n/k groups of size k and sort each piece of $O(k \log k)$ time. This preserves the property that no element is more than k element out of position. Now merge each block of k elements with the block to its left.

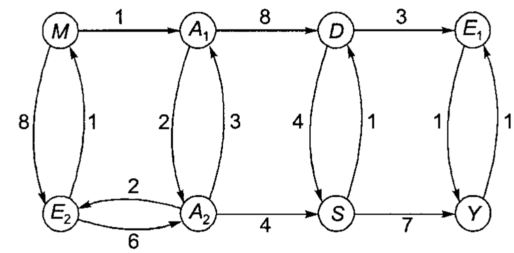
■■■

4 CHAPTER

Greedy Techniques

Q.1 Assume that letters p, q, r, s, t and u have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ and $\frac{1}{32}$ respectively. What is the average length of the message using Huffman's coding?

Q.2 Consider the following graph



Find which of the following edge is never used to compute the shortest path from single source(M) to every other vertex using dijkstra's algorithm.

(a) (M, A_1) (b) (D, E_1)
(c) (A_2, S) (d) (S, Y)

Q.3 Find the running time of dijkstra's algorithm on complete graph of n-vertices.

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

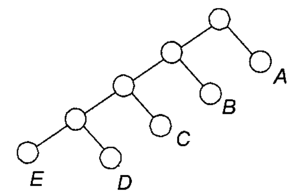
Q.4 Consider the following instance of knapsack problem.

Item	x_1	x_2	x_3	x_4	x_5
Profit	15	12	9	16	17
Weight	2	5	3	4	6

The maximum weight of 12 is allowed in the knapsack. Find the value of maximum profit with the optimal solution of the fractional knapsack problem.

(a) 31 (b) 40.2
(c) 48.5 (d) None of these

Q.5 Consider the following decode tree (Called Huffman codes) corresponding to codes for message A, B, C, D, E. The left and right branch of the tree is interpreted as 1 and 0 receptively. The distance of the external node is counted as bit sequence in decimal from the root node. Then, what will be expected decode time if message transmitted is $DBACABEED$?



(a) 9 units (b) 31 units
(c) 45 units (d) 68 units

Q.6 Which of the following is true about Huffman coding?

(a) Huffman coding may become lossy in some cases
(b) In Huffman coding no code is prefix of any other code
(c) Huffman codes may not be optimal lossless codes in some cases.
(d) All of the above

Q.7 Find the maximum possible value for fractional Knapsak problem?

Item	a	b	c	d	e	f	g	h	i	j
Weight	3	5	2	1	12	10	9	9	4	1
Value	7	10	3	3	26	19	18	17	5	4

Knapsack capacity (W) = 20.

Q.8 Given a set $A = \{A_1, A_2, ..., A_n\}$ of n activities with start and finish time (Si, fi), $1 \le i \le n$, select maximal set S of "non-overlapping" activities.If the given problem is solved using greedy algorithm, then what will be the time complexity?

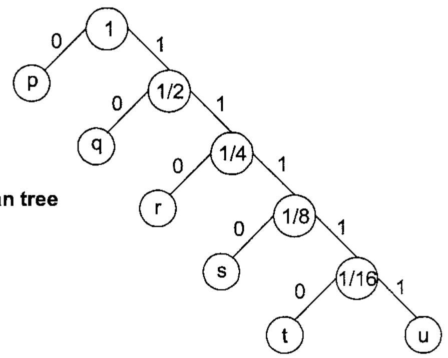
(a) $O(n^3)$ (b) $O(n^2)$
(c) $O(n \log n)$ (d) $O(n^n)$

Answers Greedy Techniques

2. (d) 3. (b) 4. (c) 5. (d) 6. (b) 8. (c)

Explanations Greedy Techniques

1. (1.9375)



Huffman tree

p : 0
q : 10
r : 110
s : 1110
t : 11110
u = 11111

average length

$$= \frac{1}{2}\times 1+\frac{1}{4}\times 2+\frac{1}{8}\times 3+\frac{1}{16}\times 4+\frac{1\times 5}{32}+\frac{1}{32}\times 5$$

$$\Rightarrow \frac{1}{2}+\frac{1}{2}+\frac{3}{8}+\frac{1}{4}+\frac{10}{32}$$

$$\Rightarrow \frac{16+16+12+8+10}{32} \Rightarrow \frac{62}{32} \Rightarrow 1.9375$$

2. (d)

M to A_1: $M \to A_1$
M to E_2 : $M \to A_1 \to A_2 \to E_2$
M to A_2: $M \to A_1 \to A_2$
M to D : $M \to A_1 \to A_2 \to S \to D$
M to S : $M \to A_1 \to A_2 \to S$
M to E_1: $M \to A_1 \to A_2 \to S \to D \to E_1$
M to Y : $M \to A_1 \to A_2 \to S \to D \to E_1 \to Y$
(S, Y) Edge not used in any shortest path.

3. (b)

Time complexity: O(E + Vlog) using fibonacci heap.

$$|V| = n$$

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

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

OR

Time complexity : O (E + V^2) using array.
Therefore

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

4. (c)

Item	x_1	x_2	x_3	x_4	x_5
P_i	15	12	9	16	17
W_i	2	5	3	4	6
P_i/W_i	7.5	2.4	3	4	2.8

Decreasing order of P_i/W_i is:

x_1, x_4, x_3, x_5, x_2

$x_1 \Rightarrow \quad P = 15, W = 2$

$\{x_1, x_4\} \Rightarrow \quad P = 15 + 16 = 31$

$W = 2 + 4 = 6$

$\{x_1, x_4, x_3\} \Rightarrow P = 31 + 9 = 40$

$W = 6 + 3 = 9$

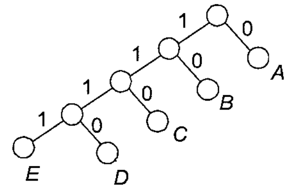
$\{x_1, x_4, x_3, x_5/2\}$

$$\Rightarrow \quad P = 40 + \frac{17}{2} = 48.5$$

$W = 9 + 6/2 = 12$

$\therefore \quad P = 48.5$

5. (d)



$A \rightarrow 0$
$B \rightarrow 10 = 2$
$C \rightarrow 110 = 6$
$D \rightarrow 1110 = 14$
$E \rightarrow 1111 = 15$

expected decode time = $\sum_{i=1}^{5} q_i d_i$

q_i = frequency of message m_i
d_i = distance from root to message m_i
the given sequence is $D\,B\,A\,C\,A\,B\,E\,E\,D$

$$\sum_{i=1}^{5} q_i d_i = 2(0 + 2 + 14 + 15) + 6 = 68 \text{ units}$$

6. **(b)**
It is never lossy and it is always optimal. The codes assigned to input characters are prefix code i.e. the codes are assigned in such a way that code assigned to one character is not prefix of code assigned to any other character.

7. **(46)**
Select all of items *a*, *d*, *e*, *j* and select 1/3 of item *g*.

$$\text{Total weight} = 3 + 1 + 12 + 1 + \frac{1}{3} \times 9 = 20$$

Total profit=7 + 3 + 26 + 4 +1/3 × 18 = 46

8. **(c)**
Sort *A* by finish time
$S = \{A_1\}$
$J = 1$
For $i = 2$ to n
Do
If $Si \geq Sj$
$S = S \cup \{Ai\}$
$J = i$
Then $\Rightarrow 0\,(n \log n)$

5 CHAPTER

Graph Based Algorithms

Q.1 The number of spanning trees of an undirected complete graph with 7 nodes is _______.

Q.2 Number of undirected graphs (not necessarily connected) can be constructed given set $V = \{1, 2, 3, 4\}$ of 4 vertices, are _______.

Q.3 Consider the following statements

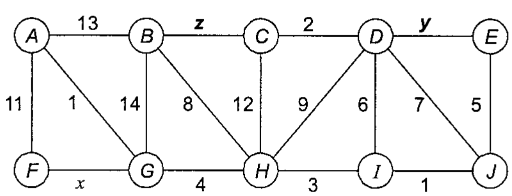
I. For every weighted graph and any two vertices s and t, Bellman-ford algorithm starting at s will always return a shortest path to t.

II. At the termination of the Bellman-ford algorithm, even if graph has negative weight cycle, a correct shortest path is found for a vertex for which shortest path is well-defined.

Which of the above statements are true?

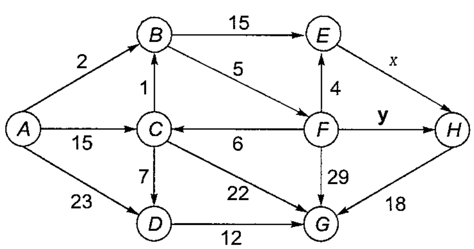
(a) only I (b) only II
(c) both I and II (d) None of these

Q.4 Suppose that minimum spanning tree of the following edge weighted graph contains the edges with weights x, y and z



What is the maximum value of $x + y + z$?

Q.5 Suppose that you are running Dijkstra's algorithm on the edge-weighted digraph below, starting from vertex A.



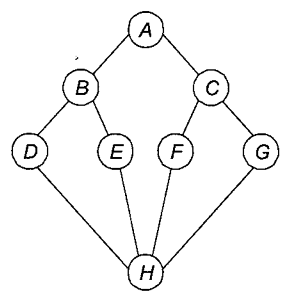
The table gives 'Distance' and 'Parent' entry of each vertex after vertex E has been deleted from the priority queue and relaxed.

Vertex	Distance	Parent
A	0	NULL
B	2	A
C	13	F
D	23	A
E	11	F
F	7	B
G	36	F
H	19	E

What could be the possible values of x and y?

(a) $x = 11$, y = 7 (b) $x = 8$, y= 12
(c) $x = 6$, y = 10 (d) $x = 10$, y = 14

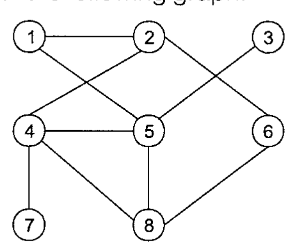
Q.6 Consider the following graph



Which of the following is NOT a depth first search traversal of the above graph?

(a) ACFHEBDG (b) ACFHGEBD
(c) ABDHFCGE (d) ABDEHFCG

Q.7 Consider the following graph:



Which does not represent depth first search (*DFS*) sequence for the above graph?

(a) 12475386 (b) 12685437
(c) 12453867 (d) 12458637

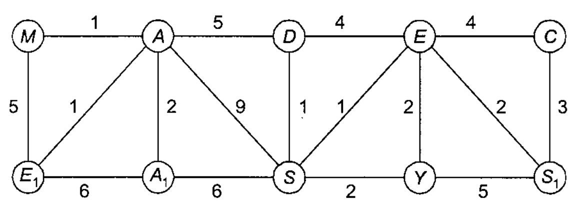
Q.8 Which of the following statements is true?

(a) Adding a constant to every edge weight in a directed graph can change the set of edges that belongs to shortest path tree. Assume unique weights.
(b) Adding a constant to every edge weight in an undirected graph can change the set of edges that belongs to the minimum spanning tree .
(c) Rerunning Dijkstra's algorithm on a graph *V* times will result in the correct shortest paths tree, even if there are negative edges (but no negative cycles).
(d) None of these

Q.9 Which of the following statements is invalid?

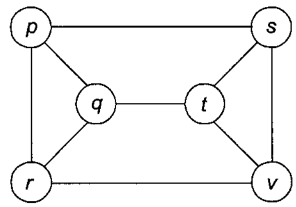
(a) A graph where all edge weights are distinct can have more than one shortest path between two vertices.
(b) Multiplying all edge weights by a positive number might change the cost of minimum spanning tree.
(c) Both (a) and (b)
(d) Neither (a) nor (b)

Q.10 Assume that *G* be a graph with (v, e) where v is the set of vertices and e is the set of edges in *G*. Assume that E_{sp} be the cost of edges of shortest path from *M* to *S*, which is computed by Dijkstra's algorithm. The graph *G* is given below.



Let E_{max} is the cost of all edges which show shortest path from *M* to every other vertices. What is the value of $E_{max} - E_{sp}$?

Q.11 Consider the following graph *G*.

G : 

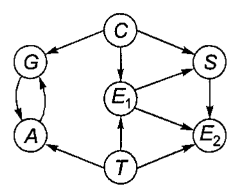
Modified DFS on G applied as follows:

- Starting vertex is '*p*'.
- Vertex is visited based on alphabetic order.
- Vertices are visited in order *p*, *q*, *r*, *s*, *t*, *v*.
- It works same as DFS except the visiting order restriction

What is the number of back edges during the above DFS traversal on *G*?

(a) 2 (b) 3
(c) 4 (d) 5

Q.12 Find the number of strong components in the following graph. Strong component is the maximum subgraph which is strongly connected. Union of all strong components should contain all vertices of given graph.



(a) 1 (b) 2
(c) 3 (d) None of these

Q.13 Consider the following statements:

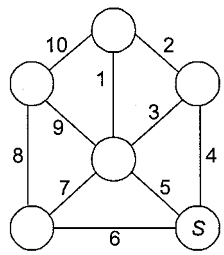
S_1: DFS of directed graph always produces the same number of edges in the traversal irrespective of starting vertex.

S_2: If we remove all of back edges found while DFS traversal on directed graph, the resultant graph is acyclic.

Find which of the following is true.

(a) Both S_1 and S_2 are valid
(b) Only S_1 valid
(c) Only S_2 Valid
(d) Neither S_1 nor S_2 is valid

Q.14 Consider the following graph.



Assume that node 'S' is the starting vertex for prim's algorithm. Which of the following can be the correct order of edges in which they are added to construct the minimal spanning tree?

(a) 4, 2, 1, 6, 8 (b) 4, 1, 2, 6, 8
(c) 4, 2, 1, 7, 10 (d) None of these

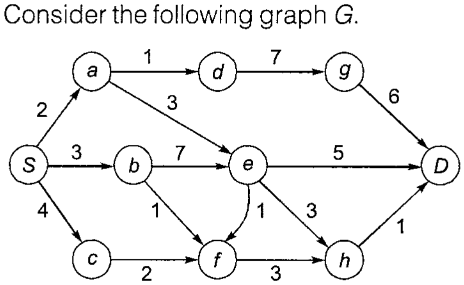
Q.15 Choose the invalid statement from the following?

(a) A graph can have more than one shortest path between two vertices.
(b) A graph where all edge weights are distinct can have more than one shortest path between two vertices.
(c) Multiplying all edge weights by a positive number might change the graph's minimum spanning tree.
(d) Both (b) and (c)

Q.16 Assume a graph G has negative weight edges. Which of the following statement is correct when the dijkstra's algorithm is applied on G?

(a) Always produces correct results
(b) Always produces incorrect results
(c) It always terminates but may produce incorrect results
(d) None of the above

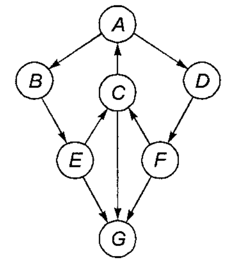
Q.17 Consider the following graph G.



What is minimum distance from S to D?

(a) 8 (b) 16
(c) 9 (d) 15

Q.18 Consider the following graph G.



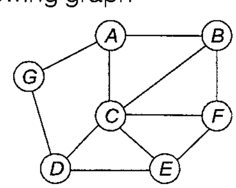
Perform a DFS on G starting at vertex A and selection of adjacent vertex in DFS decided by the lexicographical order. In graph G, B and D are adjacent to A. First it selects B, because B comes first in lexicographical order. Find number of cross edges when the given DFS is performed on G?

(a) 1 (b) 2
(c) 3 (d) 4

Q.19 Let the weight of minimum spanning tree of given graph G as per Kruskal's algorithm is w_K and as per prim's algorithm is w_P then

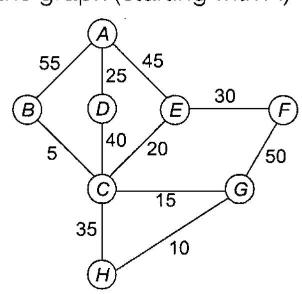
(a) $w_K \leq w_P$
(b) $w_K \geq w_P$
(c) $w_K = w_P$
(d) Any of the above is possible depends on graph structure

Q.20 Which of the following are the articulation points in the following graph



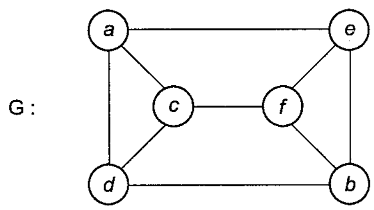
(a) Only C (b) D, C, F
(c) B, C, F (d) None of these

Q.21 Given the graph (starting with A)



What is the cost of minimum cost spanning tree.
(a) 195 (b) 150
(c) 145 (d) 140

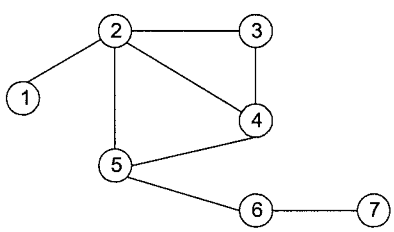
Q.22 Consider the following graph G.



Modified BFS traversal on G applied as follows
- Assume starting vertex is '*a*'.
- At any level (breadth), vertices are visited in alphabetic order.
- In above graph G, after visiting vertex '*a*', the order of vertices visited in G are *c*, *d* and *e*.
- It works same as BFS except the restriction on order of vertices visited.

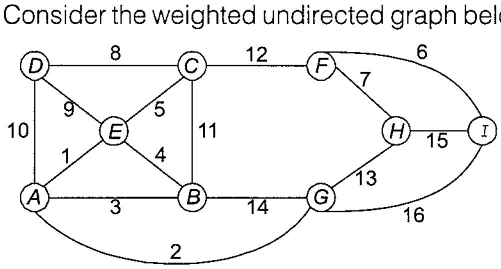
Which of the following is not a cross edge during given BFS traversal on G?
(a) {d, c} (b) {d, b}
(c) {e, b} (d) {e, f}

Q.23 Consider the following graph:



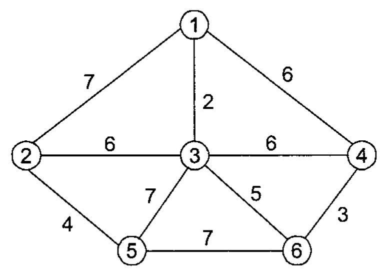
What is the depth first search (DFS) sequence for the above graph?
(a) 4, 5, 2, 1, 3, 6, 7 (b) 2, 3, 4, 5, 6, 1, 7
(c) 2, 1, 5, 3, 4, 6, 7 (d) 4, 2, 1, 5, 6, 3, 7

Q.24 Consider the weighted undirected graph below



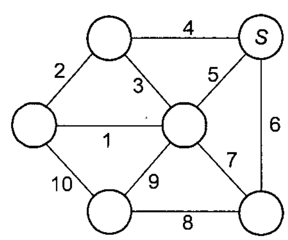
Assume prim's algorithm and kruskal's algorithm are executed on the above graph to find the minimum spanning tree. For a particular edge (e_i) which is included in minimum spanning tree and the position of an edge in minimum spanning tree is denoted by e_{p_i}. Where $1 \le e_{p_i} \le 8$ (where position defines the order in which edges are included in the MST). Then what is the maximum value of $\left|(e_{p_i})_{\text{prim's}} - (e_{p_i})_{\text{kruskal's}}\right|$?

Q.25 Consider the undirected weighted graph in Fig. The minimum cost spanning tree for this graph has the cost.



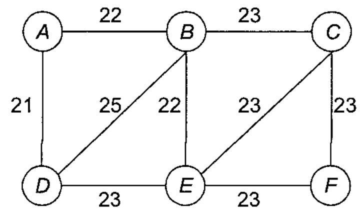
(a) 18 (b) 20
(c) 24 (d) 22

Q.26 Consider the following graph.



Assume node 'S' is the starting vertex for prim's algorithm. Which of the following can be the correct order of edges in which they are added to construct the minimal spanning tree?
(a) 4, 2, 1, 6, 8
(b) 4, 1, 2, 6, 8
(c) 4, 2, 1, 7, 10
(d) None of these

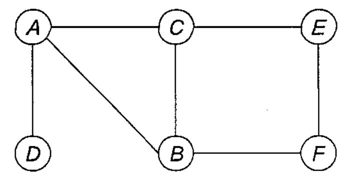
Q.27 Consider the following graph G.



Find the number of minimum cost spanning trees using kruskal's algorithm or prim's algorithm.

(a) 3 (b) 5
(c) 7 (d) 4

Q.28 The depth first search algorithm has been implemented using the stack data structure. One possible order of visiting the node of the following graph is



(a) *A D B C F E* (b) *A C E B F D*
(c) *B A C D F E* (d) *B C E F A D*

■■■■

Answers Graph Based Algorithms

3.	(b)	5.	(b)	6.	(d)	7.	(b)	8.	(a)	9.	(b)	11.	(c)	12.	(c)	13.	(c)
14.	(a)	15.	(c)	16.	(c)	17.	(a)	18.	(b)	19.	(c)	20.	(d)	21.	(c)	22.	(b)
23.	(a)	25.	(b)	26.	(a)	27.	(b)	28.	(d)								

Explanations Graph Based Algorithms

1. **(16807)**
Number of spanning trees for an undirected complete graph with nodes $n = n^{n-2}$.

2. **(64)**
As number of undirected graph is $= 2^{(n(n-1)/2)}$

3. **(b)**
I: If the graph contains a negative weight cycle then no shortest path exist.
II: If shortest path is well defined then it contains at most V-1 edges. Running V-1 iterations of Bellman ford will find the path.

4. **(25)**

$$x \le 11$$
$$y \le 6$$
$$z \le 8$$

5. **(b)**

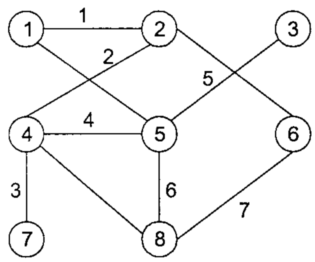
	A	B	C	D	E	F	G	H
	0_{Nill}	∞	∞	∞	∞	∞	∞	∞
A	–	2_A	15_A	23_A	∞	∞	∞	∞
B	–	–	15_A	23_A	17_B	7_B	∞	∞
F	–	–	13_F	23_A	11_F	–	36_F	$(y+7)_F$
Given → E	–	–	13_F	23_A	–	–	36_F	$(x+11)_E$

Compare above table with given table

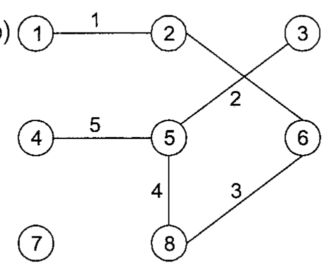
$$x + 11 = 19$$
$$x = 8$$
$$y + 7 \ge x + 11$$
$$y + 7 \ge 8 + 11$$
$$\Rightarrow \quad y \ge 12$$

6. **(d)**
AB<u>DE H</u> FGH is not DFS traversal

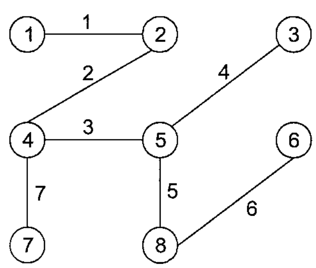
7. (b)

(a) 

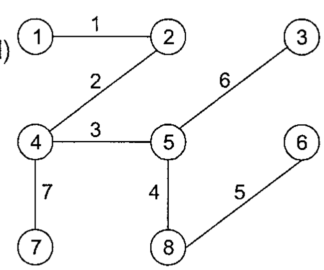
DFS safe sequence

(b) 

4 to 3 not follow DFS so not DFS sequence because have to go 4 to 7 first then 7 to 3.

(c) 

DFS safe sequence

(d) 

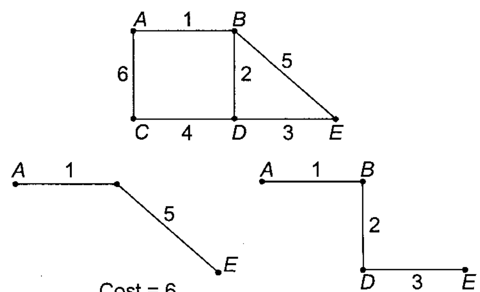
DFS safe sequence

8. (a)

Adding a constant to every edge weight in a directed graph can change the set of edges that belongs to shortest path tree. Assume unique weights.

9. (b)

(*i*) Graph can have more than one shortest path between two vertex where all edge weights are distinct.



Cost = 6 Cost = 6

(*ii*) Multiplying all edge weight by a positive number should not effect cost of minimum spanning tree.

So option (b) is invalid.

10. (51)

Minimum spanning tree by using Dijkstra's is

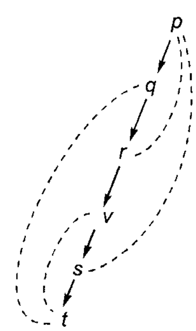
$E_{max} = 0+2+1+3+6+7+9+8+10+12 = 58$

$E_{sp} = 1 + 5 + 1 = 7$

Value of $E_{max} - E_{sp} = 58 - 7 = 51$

11. (c)

Apply the given DFS



Tree edges are: $\{p, q\}, \{q, r\}, \{r, v\}, \{v, s\}, \{s, t\}$.

Back edges are: $\{t, q\}, \{t, v\}, \{s, p\}, \{r, p\}$.

12. (c)

There are three strong components

1. G, A
2. E_1, T, S, E_2
3. C

$\therefore$ Option (c) is correct.

13. (c)

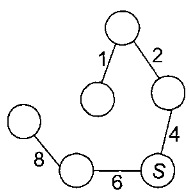
S_1: $A \rightarrow B \rightarrow C$

Starting with A will get 2 edges
Starting with B will get only 1 edge
Starting with C will get no edge
$\therefore$ DFS on directed graph may not give same number of edges.

S_2: Removing all back edges makes the graph as acyclic.
So S_2 is valid statement.

Option (c) is correct.

14. (a)



Starting vertex is S :
Selecting of edges in MST : 4, 2, 1, 6, 8
$\therefore$ Option (a) is correct.

15. (c)

Multiplying all edge weights by a positive number does not change the graph's minimum spanning tree.
$\therefore$ Option (c) is not valid.

16. (c)

Dijkstra's algorithm always terminates after $|E|$ relaxations and $|V| + |E|$ priority queue operations, but may produce incorrect results.
So, option (c) is correct.

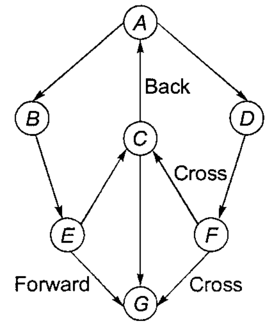
17. (a)

$S \xrightarrow[3]{} b \xrightarrow[1]{} f \xrightarrow[3]{} h \xrightarrow[1]{} D$

Minimum distance = 3 + 1 + 3 + 1 = 8

18. (b)

$\Rightarrow$ 2 Cross edges in G,
1 Forward edge in G,
1 Back edge in G
$\therefore$ # cross edges = 2



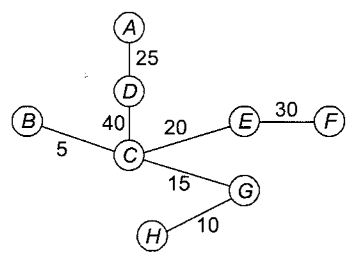
19. (c)

There will be unique minimum cost for any minimum spanning tree in a graph irrespective of algorithm applied. i.e. more than one minimum cost spanning trees are possible but their cost will be same.

20. (d)

In a connected graph, a vertex v is said be an articulation point if by removing that vertex together with its edges the graph become disconnected. In the given graph there is no articulation point.

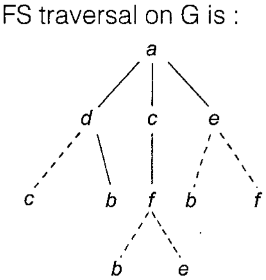
21. (c)



5 + 10 + 15 + 20 + 25 + 30 + 40 = 145

22. (b)

Given BFS traversal on G is :



Tree edges are:{a, d}, {a, c}, {a, e}, {d, b}, {c, f}
Cross edges are:{d, c}, {e, b}, {e, f}, {f, b},{f,e }
$\Rightarrow$ {d, b} is not a cross edge.

24. (2)

Kruskal's algorithm:

AE, AG, AB, CE, FI, FH, CD, CF

Prim's algorithm:

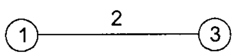
AE, AG, AB, CE, CD, CF, FI, FH

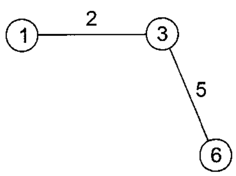
$$\max\left|(e_{p_i})_{\text{prim's}} - (e_{p_i})_{\text{kruskal's}}\right| = |5-7| = 2$$

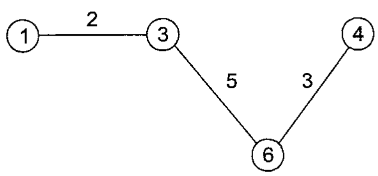
25. (b)

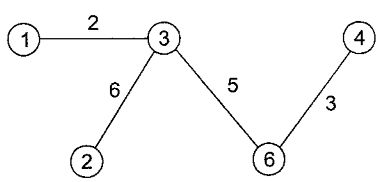
Using the Kruskal's algorithm

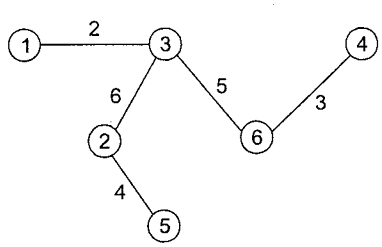
(i) 

(ii) 

(iii) 

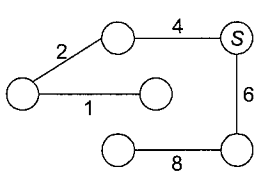
(iv) 

(v) 

(vi) 

∴ Minimum cost is 20.

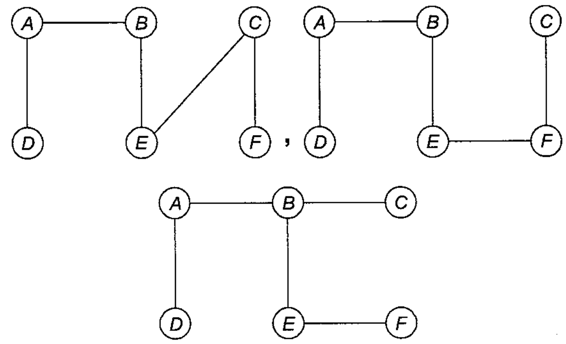
26. (a)

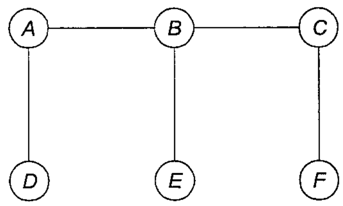


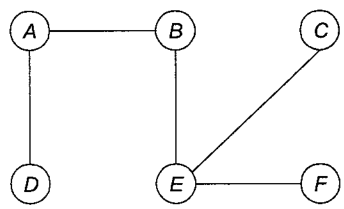
Starting vertex is S :

Selecting of edges in MST : 4, 2, 1, 6, 8

27. (b)

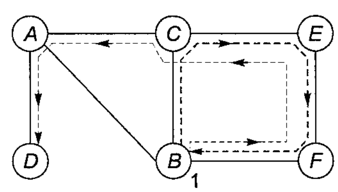






All above trees MST with cost: 21 + 22 + 22 + 23 + 23 = 111

28. (d)



∵ $BCEFAD$ is correct depth first search order.

■■■■

6 CHAPTER Dynamic Programing

Q.1 Consider the following problems.

1. Longest common subsequence
2. Optimal Binary search tree
3. 0/1 knapsack problem
4. Matrix chain multiplication

Which of the above problems can be solved using dynamic programming?

(a) 1 and 2 only (b) 2 and 3 only
(c) 1, 3 and 4 only (d) 1, 2, 3 and 4

Q.2 Consider the following knapsack problem with capacity $W = 8$

Item	Profit	Weight
I_1	13	1
I_2	8	5
I_3	7	3
I_4	3	4

Which of the following item is not selected in the optimal solution of 0/1 knapsack problem?

(a) I_1 only (b) I_2 only
(c) I_3 only (d) None of these

Q.3 Consider the following strings x and y.

x = "csemadeeasy"
y = "gateexam"

How many longest common subsequences are possible from x and y?

Q.4 Given two sequences A and B. Let $X(A, B)$ denote the number of times that A appears as subsequence of B i.e. sequence ab appears 4 times as a subsequence of aebabdb. Let A_i denotes the first i characters of string A and $A[i]$ denote the i^{th} character.

Which of the following will computes the recurrence relation $C(A_i, B_j)$?

$$C(A_i, B_j) = \begin{cases} 1; & \text{if } (i = 0) \\ 0; & \text{if } (i > 0 \text{ and } j = 0) \\ \text{----} & \text{--------------} \\ \text{----} & \text{--------------} \end{cases}$$

(a) $C(A_i, B_{j-1})$; if $A[i] \neq B[j]$
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] = B[j]$

(b) $C(A_{i-1}, B_j)$; if $A[i] = B[j]$
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] = B[j]$

(c) $C(A_i, B_{j-1})$; if $A[i] = B[j]$
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] \neq B[j]$

(d) None of the above

Q.5 Consider the following matrices with dimensions.

A_1 is 4×6
A_2 is 6×8
A_3 is 8×4
A_4 is 4×5

Which of the following multiplication order gives optimal solution?

(a) $((A_1 A_2) A_3) A_4$ (b) $(A_1 (A_2 A_3))A_4$
(c) $A_1 ((A_2 A_3) A_4)$ (d) $(A_1 A_2) (A_3 A_4)$

Q.6 The difference between maximum possible profit for 0/1 Knapsack and fractional Knapsack problem with capacity $(W) = 20$.

Item	a	b	c	d	e	f	g	h	i	j
Weight	3	5	2	1	12	10	9	9	4	1
Profit	7	10	3	3	26	19	18	17	5	4

Q.7 Consider the following adjacency matrix 'D' for represented the directed graph with distances (costs) between every pair of vertices.

$$D = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{matrix} A & B & C \end{matrix} \\ \begin{bmatrix} 0 & 3 & 2 \\ 1 & 0 & 2 \\ 4 & 1 & 0 \end{bmatrix} \end{array}$$

If warshall's algorithm is applied on 'D' to compute the shortest distances then following resultant matrix is obtained.

$$D^* = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0 & 3 & 2 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix} \end{array}$$

If the edge from B to C in the given directed graph is removed then how many shortest distances will be changed in D^*?

(a) 3 (b) 4
(c) 5 (d) 6

Q.8 What is the running time of efficient algorithm for the finding the shortest path between two vertices in a directed graph? Assume that all edges are having equal weights. (V is set of vertices and E is set of Edges)

(a) $|V|$ (b) $|V| \log |E|$
(c) $O(|V| + |E|)$ (d) None of these

Q.9 Match the following groups

Group-1
- **A.** Divide and conquer
- **B.** Greedy approach
- **C.** Dynamic programming
- **D.** Back tracking

Group-2
1. Longest common subsequence
2. Huffman codes
3. Quick sort
4. Chess game

Codes:

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

(d) None of these

Q.10 Let $X = < x_1, x_2, \ldots x_m >$ and $Y = < y_1, y_2, \ldots y_n >$ be sequences. What is the running time to find the longest common subsequence of X and Y using dynamic programming.

(a) $O(m + n)$ (b) $O(mn)$
(c) $O(m^n)$ (d) $O(n^m)$

Q.11 Let X = abcbdab and Y = *bdcaba*.
Find the length of longest common subsequence (LCS) of X and Y.

(a) 3 (b) 4
(c) 5 (d) 6

Q.12 Let $G = (V, E)$ be a directed graph. Each edge of G is represented as (i, j) with length $l[i, j]$. If there is no edge from i to j then $l[i, j] = \infty$. Assume n vertices in V and $d^k_{i,j}$ is the length of shortest path from i to j that does not pass through any vertex in $\{k+1, k+2, \ldots, n\}$.

$$d^k_{ij} = \begin{cases} l[i, j] & \text{if } k = 0 \\ \min\{A, B\} & \text{if } 1 \le k \le n \end{cases}$$

If the above $d^k_{i,j}$ computed recursively to find all pairs shortest path, identify A and B respectively?

(a) $d^{k-1}_{i,j}$ and $d^{j-1}_{i,j} + d^{j-1}_{k,j}$

(b) $d^{k-1}_{i,j}$ and $d^{k-1}_{i,k} + d^{k-1}_{k,j}$

(c) $d^k_{i,j}$ and $d^k_{i,k} + d^k_{k,j}$

(d) $d^k_{i,j}$ and $d^k_{i,k} + d^k_{j,k}$

Q.13 Given a set of n points: $S = \{P_1, P_2, P_3, P_4 \ldots P_n\}$, where $P_i = (x_i, y_i)$. Finding the pair of points that has the smallest distance among all pairs that can be solved in O(nlogn) time using

(a) Divide and Conquer
(b) Greedy Technique
(c) Dynamic programming
(d) All of these

Q.14 Let $P_0 = 5$, $P_1 = 6$, $P_2 = 7$, $P_3 = 1$, $P_4 = 10$, $P_5 = 2$ and for $1 \le i \le 5$. Let X_i be a matrix with P_{i-1} rows and P_i columns, and $X = X_1.X_2.X_3.X_4.X_5$. Which of the following is optimum parenthesization for computing X?

(a) $((X_1(X_2 \cdot X_3))X_4)X_5$
(b) $(X_1(X_2 \cdot X_3))(X_4 \cdot X_5)$
(c) $(X_1X_2)((X_3X_4) \cdot X_5)$
(d) $((X_1 \cdot X_2)(X_3 \cdot X_4))X_5$

Q.15 How many number of longest common subsequences of X = gate123 and Y = aget 321 can be formed?

(a) 2 (b) 8
(c) 12 (d) 14

Q.16 Consider the following statements.

S_1 : Greedy algorithm exists

S_2 : Dynamic programming algorithm exists

S_3 : Exhibit optimal substructure property.

Which of the above statements are true for 0-1 knapsack problem?

(a) Only S_1 and S_2 (b) Only S_2 and S_3

(c) Only S_1 and S_3 (d) S_1, S_2 and S_3

Q.17 Match **List-I** with **List-II** and select the correct answer using the codes given below the lists:

List-I	List-II
A. $O(n^2)$	1. Merge sort
B. $O(n)$	2. The towers of Hanoi problem
C. $O(n \log n)$	3. Shortest path between two nodes in graph
D. $O(2^n)$	4. Finding an element in the unsorted array

Codes:

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

Q.18 Given an array of n numbers, give an algorithm for finding a contiguous subsequence $A(i) \dots A(j)$ for which the sum of elements is maximum

Eg. $[-2, 11, -4, 13, -5, 2] \rightarrow 20$

If dynamic programming approach is used then what is time complexity and space complexity?

(a) $O(n^3), O(1)$ (b) $O(n), O(n)$

(c) $O(n^3), O(n)$ (d) $O(n^2), O(1)$

Answers Dynamic Programing

1.	(d)	2.	(b)	4.	(a)	5.	(b)	8.	(c)	9.	(c)	10.	(b)
11.	(b)	12.	(b)	13.	(a)	14.	(b)	15.	(c)	16.	(b)	17.	(b)
18.	(b)												

Explanations Dynamic Programing

1. (d)

Longest common subsequence, longest increasing subsequence, sum of subsets, optimal BST, 0/1 knapsack, matrix chain multiplication, Travelling salesperson, Balanced partition, Fibonacci sequence, Multistage graph problems are solved using dynamic programming.

2. (b)

$$W = 8 \text{ (capacity)}$$

Feasible solutions:

(i) $\{I_1, I_3, I_4\}$,

(ii) $\{I_2, I_3\}$

Profit of $\{I_1, I_3, I_4\}$ is 23

profit of $\{I_2, I_3\}$ is 15

Optimal solution is $\{I_1, I_3, I_4\}$ with capacity of 8 and maximum profit 23 produced.

$\therefore I_2$ is not selected in the solution.

3. (1)

$$x = \text{csemadeeasy}$$
$$y = \text{gateexam}$$
$$\text{LCS}(x, y) = \text{aeea}$$

Only "aeea" is longest common subsequence.

4. (a)

$$C(A_i, B_j) = \begin{cases} 1; & \text{if } (i = 0) \\ 0; & \text{if } (i > 0 \text{ and } j = 0) \\ C(A_i, B_{j-1}); & \text{if } (A[i] \neq B[j]) \text{ when elements are not matched.} \\ C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1}); & \text{if } (A[i] = B[j]) \text{ when elements are matched.} \end{cases}$$

5. (b)

(a) $((A_1 A_2)A_3) A_4$ requires $(4 \times 6 \times 8 + 4 \times 8 \times 4 + 4 \times 4 \times 5) = 400$ multiplications.

(b) $(A_1 (A_2 A_3))A_4$ requires $(6 \times 8 \times 4 + 4 \times 6 \times 4 + 4 \times 4 \times 5) = 368$ multiplications.

(c) $A_1((A_2A_3)A_4)$ requires $(6\times8\times4+6\times4\times5+4\times6\times5) = 432$ multiplications.

(d) $(A_1A_2)(A_3A_4)$ requires $(4\times6\times8+8\times4\times5+4\times8\times5) = 512$ multiplications.

6. (3)

Fractional Knapsack problem:

Select all of item '*a*', '*d*', '*e*', '*j*' and 1/3 of item 'g'

Total weight = $3+1+12+1+1/3\times9 = 20$

Total profit = $7+3+26+4+1/3\times18 = 46$

0/1 Knapsack problem:

Select all of item *j*, *d*, *a*, *e* and *c*.

Total weight = $2+1+1+3+12 = 19$

Total profit = $7+3+26+4+3 = 43$

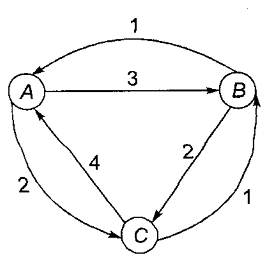
Difference = Total profit (using fractional Knapsack – Using 0/1 Knapsack)

= $46-43 = 3$

7. (1)

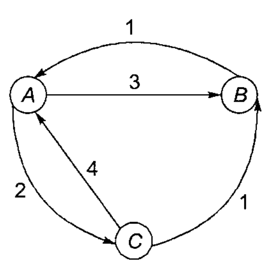
$$D = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 3 & 2 \\ B & 1 & 0 & 2 \\ C & 4 & 1 & 0 \end{array}$$

$\Rightarrow$



$$D^* = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 3 & 2 \\ B & 1 & 0 & 2 \\ C & 2 & 1 & 0 \end{array}$$

After removal of (*B*, *C*) edge, the graph is:



$$\Rightarrow D_1 = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 3 & 2 \\ B & 1 & 0 & \infty \\ C & 2 & 1 & 0 \end{array} \Rightarrow D^*_1 = \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 3 & 1 \\ B & 1 & 0 & 3 \\ C & 2 & 1 & 0 \end{array}$$

$\Rightarrow$ *B* to *C* shortest path distances are changed.

8. (c)

Using BFS by treating all edges as unweighted, it takes O(|V|+|E|) time.

9. (c)

Divide and conquer: Merge sort, Quick sort, Binary search, etc.

Greedy approach: Huffman codes, Dijkstra's, Kruskal's Algorithm, etc.

Dynamic programming: Longest common subsequence, All pairs shortest Paths, etc.

Backtracking: Chess game, 8-queens, Checkers, etc.

10. (b)

Time complexity of LCS = O(mn)

LCS computed as following and trace back later.

$$c[i, j] = \begin{cases} 0 & , \text{ if } i=0 \text{ or } j=0 \\ c[i-1, j-1]+1 & , \text{ if } x_i = y_j, i, j>0 \\ \max c[i, j-1], c[i-1, j] & , \text{ if } x_i \neq y_j, i, j>0 \end{cases}$$

11. (b)

bcba is one possible LCS of *X* and *Y*

LCS can be computed using dynamic programming.

12. (b)

$$\begin{cases} l[i, j], & \text{if } k=0 \\ \min\left\{d_{i,j}^{k-1}, d_{i,k}^{k-1}+d_{k,j}^{k-1}\right\}, & \text{if } 1\le k\le n \end{cases}$$

$\therefore$ option (b) is correct.

13. (a)

Finding closest pairs can be solved in O(n log*n*) time using divide and conquer approach.

$T(n) = 2T(n/2) + O(n)$

$= O(n\log n)$

Algorithm:

Divide and Conquer:

Step 1: Divide the set into two equal sized parts by the line l, and recursively compute the distance in each part. [d_1 = closestpair (left half); d_2 = closestpair (right half);] and returning the points in each set in sorted order by y-coordinate.

Merge Procedure:

[$O(1)$] Step 2: Let 'd' be the minimal of two minimal distances.

$$d = \min(d_1, d_2);$$

[$O(n)$] Step 3: Eliminate points that lie farther than 'd' apart from l.

[$O(n)$] Step 4: Merge the two sorted lists into one sorted list.

[$O(n)$] Step 5: Scan the remaining points in the y-order and compute the distances of each point to its 5 neighbors.

[$O(1)$] Step 6: If any of these distances is less than 'd' then update 'd'.

14. (b)

$(X_1(X_2.X_3))(X_4.X_5)$

multiplications = 102, which is minimum compared to other options.

$$\left.\begin{array}{l} X_2.X_3 \Rightarrow \text{\# Multiplications} = 42 \\ X_4.X_5 \Rightarrow \text{\# Multiplications} = 20 \\ X_1.(X_2.X_3) \Rightarrow \text{\# Multiplications} = 30 \\ (X_1.(X_2.X_3))_{5\times 1}(X_4.X_5)_{1\times 2} \Rightarrow \text{\# Multiplications} = 10 \end{array}\right\}$$

$\Rightarrow$ 102 Multiplications

15. (c)

$$X = gate123$$
$$Y = aget321$$

gt1, gt2, gt3, at1, at2, at3, ge1, ge2, ge3, ae1, ae2, ae3

Length of LCS = 3

Number of LCS's with length 3 are 12

16. (b)

No greedy algorithm exists for 0-1 knapsack problem. 0-1 knapsack problem exhibit optimal substructure property and it has only dynamic programming algorithm.

18. (b)

Initialize:

Till_max = 0;
Upto_max = 0;

Loop for each element of the array

(a) Upto_max = Upto_max +a[i];
(b) If(upto_max < 0)
    Upto_max = 0;
(c) If(Till_max < Upto_max)
    Till_max = Upto_max; return Till_max;
    It takes $O(n)$ time complexity and $O(n)$ space complexity in the form of table.

■■■■