

For 100% Result Oriented IGNOU Coaching and Project Training

Call CPD: 011-65164822, 08860352748

ASSIGNMENT OF MCS-031(ADA)

Question 1:

(a) Show, through appropriate examples or otherwise, that the following three characteristics of an algorithm are independent

(i) finiteness

An algorithm must terminate after a finite number of steps and further each step must be executable in finite amount of time. In order to establish a sequence of steps as an algorithm, it should be established that it terminates (in finite number of steps) on all allowed inputs.

(ii) definiteness

Each steps of an algorithm must be precisely defined; the action to be carried out must be rigorously and unambiguously specified for each case.

(iii) effectiveness

it consists of basic instructions that are realizable. This means that the instructions can be performed by using the given inputs in a finite amount of time.

(c) Arrange the following growth rates in increasing order: $O((24)^n)$, $O(n^2)$, $O(1)$, $O(n \log n)$

ANS-

Order Of Notation In Increasing Order:-

$O(1)$
 $O(n)^{1/2}$
 $o(n)$
 $o(n \log n)$
 $o(n^2)$
 $o(n^3)$
 $o(2^n)$

order is

$$o(1) < o(n \log n) < o(n^2) < o(2^n)$$

**For 100% Result Oriented IGNOU Coaching and Project
Training**

Call CPD: 011-65164822, 08860352748

(b) Let $f(n)$ denote the n th term of a sequence of integers given (5 marks)
by the equation $f(n) = f(n-1) + f(n-2)$ for $n > 2$ and $f(1) = 1$
and $f(2) = 1$, then using principle of mathematical induction, show
that $f(n) = \frac{(1 + \sqrt{5})^n - (1 - \sqrt{5})^n}{2^n}$ for all $n \geq 1$

**For 100% Result Oriented IGNOU Coaching and Project
Training**

Call CPD: 011-65164822, 08860352748

ANS-

CPD

For 100% Result Oriented IGNOU Coaching and Project Training

Call CPD: 011-65164822, 08860352748

The Fibonacci numbers, f_0, f_1, f_2, \dots , are defined recursively by the equations $f_0 = 0, f_1 = 1$, and

$$f_n = f_{n-1} + f_{n-2}$$

for $n = 2, 3, 4, \dots$. Fibonacci numbers form a sequence $0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$. Using Strong Induction we can prove that

$$f_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n$$

for $n = 0, 1, 2, \dots$.

Proof: Let $P(n)$ be the statement that this formula is true, where $n \geq 0$.

(Base Case) We first show $P(0)$ and $P(1)$ are true, i.e. that the formula is valid when $n = 0$ and $n = 1$. If $n = 0$, then

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^0 - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^0 = \frac{1}{\sqrt{5}} - \frac{1}{\sqrt{5}} = 0 = f_0$$

and if $n = 1$, then

$$\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^1 - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^1 = \frac{1+\sqrt{5}}{2\sqrt{5}} - \frac{1-\sqrt{5}}{2\sqrt{5}} = \frac{2\sqrt{5}}{2\sqrt{5}} = 1 = f_1$$

(Inductive Step) Let $k \geq 1$ and suppose $P(j)$ is true for all $0 \leq j \leq k$. Since $k \geq 1$, then $k-1 \geq 0$ and so in particular we are assuming that $P(k)$ and $P(k-1)$ are both true. We must show $P(k+1)$ is true. In the following we will make use of the observation that

$$\left(\frac{1 \pm \sqrt{5}}{2} \right)^2 = \frac{1 \pm 2\sqrt{5} + 5}{4} = \frac{6 \pm 2\sqrt{5}}{4} = \frac{3 \pm \sqrt{5}}{2} = \frac{1 \pm \sqrt{5}}{2} + 1$$

From the recursive definition of the Fibonacci numbers and from our inductive hypothesis we get

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \\ &= \left[\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^k - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^k \right] + \left[\frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{k-1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \right] \\ &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k + \left(\frac{1+\sqrt{5}}{2} \right)^{k-1} - \left(\frac{1-\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \right] \\ &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k-1} \left(\frac{1+\sqrt{5}}{2} + 1 \right) - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \left(\frac{1-\sqrt{5}}{2} + 1 \right) \right] \\ &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{k-1} \left(\frac{1+\sqrt{5}}{2} \right)^2 - \left(\frac{1-\sqrt{5}}{2} \right)^{k-1} \left(\frac{1-\sqrt{5}}{2} \right)^2 \right] \\ &= \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{k+1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{k+1} \end{aligned}$$

Thus $P(k+1)$ is true. By Strong Induction we have proven that the formula is valid for all $n \geq 0$.

For 100% Result Oriented IGNOU Coaching and Project Training

Call CPD: 011-65164822, 08860352748

Question 2:

Suppose that instead of binary or decimal representation of integers, we have ternary, along with 3's complement representation of integers, i.e., integers are represented using three digits, viz., 0, 1, 2. For example, the integer 47 is represented as 01202 = (in decimal) $1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0$, where, the leading zero indicates positive sign. And the integer (-47) in 3's complement is represented by 11021, the leading 1 indicates negative sign. The other digits, except the right-most, in the representation of (-47) are obtained by subtracting from 2 the corresponding digit in 47's representation, and then adding 1 (the representation of -47 is obtained as $11020 + 00001$). Write a program for the arithmetic (negation of an integer, addition, subtraction and multiplication of two integers) of 4 integers using 3's complement representation. The program should include a procedure for calculating each of negation of an integer, addition, subtraction and multiplication of two integers. The integers will use 8-ternary digit positions, in which the left-most position will be used for sign. Using your program find the ternary representation of each of the decimal numbers 345, $--297$, 18 and $((345 -- 297) * 18)$ ans-

Question 3:

(a) Write a short note on each of the following:

(i) Best case analysis

In the best case analysis, we calculate lower bound on running time of an algorithm. We must know the case that causes minimum number of operations to be executed. In the linear search problem, the best case occurs when x is present at the first location. The number of operations in worst case is constant (not dependent on n). So time complexity

$O(1)$.
in the best case would be

(ii) amortized analysis

This term refers to the analysis of algorithm performance based on a specific sequence of operations that targets the *worst case scenario* -- that is, amortized analysis does imply that the metric is worst case performance (although it still does not say which quantity is being measured). To perform this analysis, we need to specify the *size* of the input, but we do not need to make any assumptions about its form.

In layman's terms, amortized analysis is picking an arbitrary size for the input and then "playing through" the algorithm. Whenever a decision that depends on the input must be

**For 100% Result Oriented IGNOU Coaching and Project
Training**

Call CPD: 011-65164822, 08860352748

made, the worst path is taken¹. After the algorithm has run to completion we divide the calculated complexity by the size of the input to produce the final result.

(b) Using one-by-one (i) bubble sort (ii) heap sort and (iii) quick sort, sort the following sequence in increasing order and analyze (i.e., find number of comparisons and assignments in each of) the algorithm: 43, 28, 37, 18, 15, 40, 12 ans-

For 100% Result Oriented IGNOU Coaching and Project
Training
Call CPD: 011-65164822, 08860352748

(B) Bubble sort

48, 28, 37, 18, 15, 40, 12

$n = 7$

Iteration-1 $i = 1$ to 6

Step1 $j = 1$ to 6

$j = 1$ $48 > 28$ True \leftrightarrow

28, 48, 37, 18, 15, 40, 12

$j = 2$ $48 > 37$ True \leftrightarrow

28, 37, 48, 18, 15, 40, 12

$j = 3$ $48 > 18$ True \leftrightarrow

28, 37, 18, 48, 15, 40, 12

$j = 4$ $48 > 15$ True \leftrightarrow

28, 37, 18, 15, 48, 40, 12

$j = 5$ $48 > 40$ True \leftrightarrow

28, 37, 18, 15, 40, 48, 12

$j = 6$ $48 > 12$ True \leftrightarrow

28, 37, 18, 15, 40, 12, 48

Comparison = 6

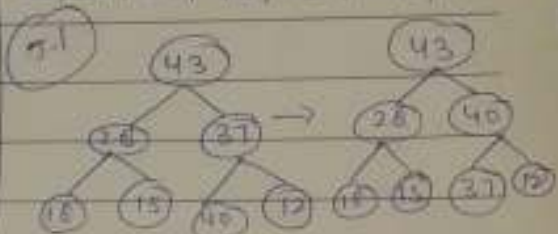
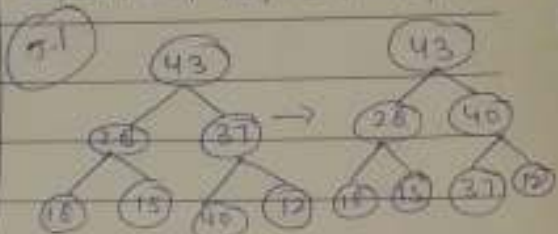
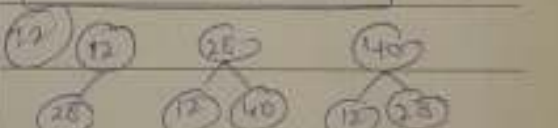
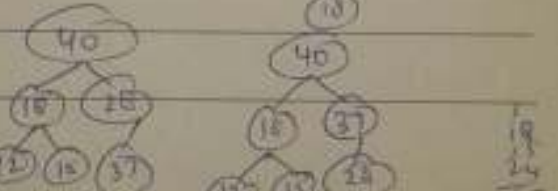
Assignment = 18

For 100% Result Oriented IGNOU Coaching and Project Training
Call CPD: 011-65164822, 08860352748

A CHOICE FOR LIFE

<p><u>Iteration-2</u> $i=2$ to 6</p> <p>$j=1$ to 5</p> <p>28, 37, 18, 15, 40, 12, 48</p> <p>No - 28, 37, 18, 15, 40, 12, 48</p> <p>\leftrightarrow - 28, 18, 37, 15, 40, 12, 48</p> <p>\leftrightarrow - 28, 18, 15, 37, 40, 12, 48</p> <p>No - 28, 18, 15, 37, 40, 12, 48</p> <p>\leftrightarrow - 28, 18, 15, 37, 12, 40, 48</p> <p>Comparison = 5</p> <p>Assignment = 15</p>	<p><u>Iteration-3</u> $i=4$ to 6</p> <p>$j=1$ to 3</p> <p>18, 15, 28, 12, 37, 40, 48</p> <p>\leftrightarrow - 15, 18, 28, 12, 37, 40, 48</p> <p>No - 15, 18, 28, 12, 37, 40, 48</p> <p>\leftrightarrow - 15, 18, 12, 28, 37, 40, 48</p> <p>Comparison = 3</p> <p>Assignment = 6</p>
<p><u>Iteration-3</u> $i=3$ to 6</p> <p>$j=1$ to 4</p> <p>28, 18, 15, 37, 12, 40, 48</p> <p>\leftrightarrow - 18, 28, 15, 37, 12, 40, 48</p> <p>\leftrightarrow - 18, 15, 28, 37, 12, 40, 48</p> <p>No - 18, 15, 28, 37, 12, 40, 48</p> <p>\leftrightarrow - 18, 15, 28, 12, 37, 40, 48</p> <p>Comparison = 4</p> <p>Assignment = 12</p>	<p><u>Iteration-4</u> $i=5$ to 6</p> <p>$j=1$ to 2</p> <p>15, 18, 12, 28, 37, 40, 48</p> <p>No - 15, 18, 12, 28, 37, 40, 48</p> <p>\leftrightarrow - 15, 12, 18, 28, 37, 40, 48</p> <p>Comparison = 2</p> <p>Assignment = 3</p>
<p><u>Iteration-5</u> $i=6$ to 6</p> <p>$j=1$ to 1</p> <p>15, 12, 18, 28, 37, 40, 48</p> <p>\leftrightarrow - 12, 15, 18, 28, 37, 40, 48</p> <p>Comparison = 1</p> <p>Assignment = 3</p>	<p>Total Comparison = 6 + 5 + 4 + 3 + 2 + 1 = 21</p> <p>Total Assignment = 18 + 15 + 12 + 6 + 3 + 3 = 57</p>

For 100% Result Oriented IGNOU Coaching and Project Training
Call CPD: 011-65164822, 08860352748

<p>I-1 Quick sort</p> <p>43, 28, 37, 18, 15, 40, 12</p> <p>Pivot = 43</p> <p>43, 28, 37, 18, 15, 40, 12</p> <p>Pivot \leftrightarrow A[3]</p> <p>12, 28, 37, 18, 15, 40, 43</p>	<p>Pivot \leftrightarrow A[3]</p> <p>12, 15, 18, 28, 37, 40, 43</p> <p>After Quick Sort</p> <p>12, 15, 18, 28, 37, 40, 43</p>
<p>I-2</p> <p>Pivot = 12</p> <p>12, 28, 37, 18, 15, 40, 43</p> <p>Pivot \leftrightarrow A[0]</p> <p>12, 28, 37, 18, 15, 40, 43</p>	<p><u>Heap Sort</u></p> <p>43, 28, 37, 18, 15, 40, 12</p> <p>I-1</p> 
<p>I-3</p> <p>Pivot = 28</p> <p>12, 28, 37, 18, 15, 40, 43</p> <p>12, 28, 15, 18, 37, 40, 43</p> <p>Pivot \leftrightarrow A[3]</p> <p>12, 18, 15, 28, 37, 40, 43</p>	 <p>43, 28, 40, 18, 15, 37, 12</p> <p>12, 28, 40, 18, 15, 37, 43</p> 
<p>I-4 Pivot = 18 Pivot = 37</p> <p>12, 18, 15 37, 40</p>	

For 100% Result Oriented IGNOU Coaching and Project Training
Call CPD: 011-65164822, 08860352748

(WONCOR 2018)
A CHOICE FOR LIFE

I-2

40	18	37	12	15	28	43
----	----	----	----	----	----	----

7-6
 15
 12

28	18	37	12	15	40	43
----	----	----	----	----	----	----

15	12	18	20	37	40	43
----	----	----	----	----	----	----

12, 15, 18, 20, 37, 40, 43

I-3

```

    graph TD
      28((28)) --- 18((18))
      28 --- 37((37))
      37 --- 18
      37 --- 20((20))
      20 --- 12((12))
      20 --- 15((15))
    
```

37	18	28	12	15	40	43
----	----	----	----	----	----	----

I-4

```

    graph TD
      15((15)) --- 18((18))
      18 --- 15
      18 --- 28((28))
      28 --- 15
      28 --- 18
      15 --- 12((12))
    
```

28	15	18	12	37	40	43
----	----	----	----	----	----	----

12	15	18	28	37	40	43
----	----	----	----	----	----	----

7-5

```

    graph TD
      12((12)) --- 15((15))
      15 --- 12
      15 --- 18((18))
      18 --- 12
      18 --- 15
    
```

18	12	15	28	37	40	43
----	----	----	----	----	----	----

15	12	18	28	37	40	43
----	----	----	----	----	----	----

Question 4:

For 100% Result Oriented IGNOU Coaching and Project Training

Call CPD: 011-65164822, 08860352748

(a) The following pseudo-code is given to compute $(a^b) \bmod n$, where a , b and n are positive integers. Trace the algorithm to compute $7^{560} \bmod 561$

MODULAR-EXPONENTION (a, b, n)

1. $c \leftarrow 0$ { variable c is assigned 0 }
2. $d \leftarrow 1$
3. *let $(b_k, b_{k-1}, \dots, B_0)$ be binary representation of b*
4. for $i \leftarrow k$ downto 0
5. do $c \leftarrow 2c$
6. $d \leftarrow (d \cdot d) \bmod n$
7. if $b_i = 1$
8. then $c \leftarrow c+1$
9. $d \leftarrow (d \cdot a) \bmod n$
10. end-do
11. return d .

(b) Explain the essential idea of Dynamic Programming. How does Dynamic Programming differ from Divide and conquer approach for solving problems?

Ans

Dynamic Programming is a technique for solving problems with overlapping subproblems. Each sub-problem is solved only once and the result of each sub-problem is stored in a table (generally implemented as an array or a hash table) for future references. These sub-solutions may be used to obtain the original solution and the technique of storing the sub-problem solutions is known as memoization.

Dynamic Programming

Breaks the problem down into discrete subproblems. The recursive algorithm for the Fibonacci sequence is an example of Dynamic Programming, because it solves for $\text{fib}(n)$ by first solving for $\text{fib}(n-1)$. In order to solve the original problem, it solves a different problem.

Divide and Conquer

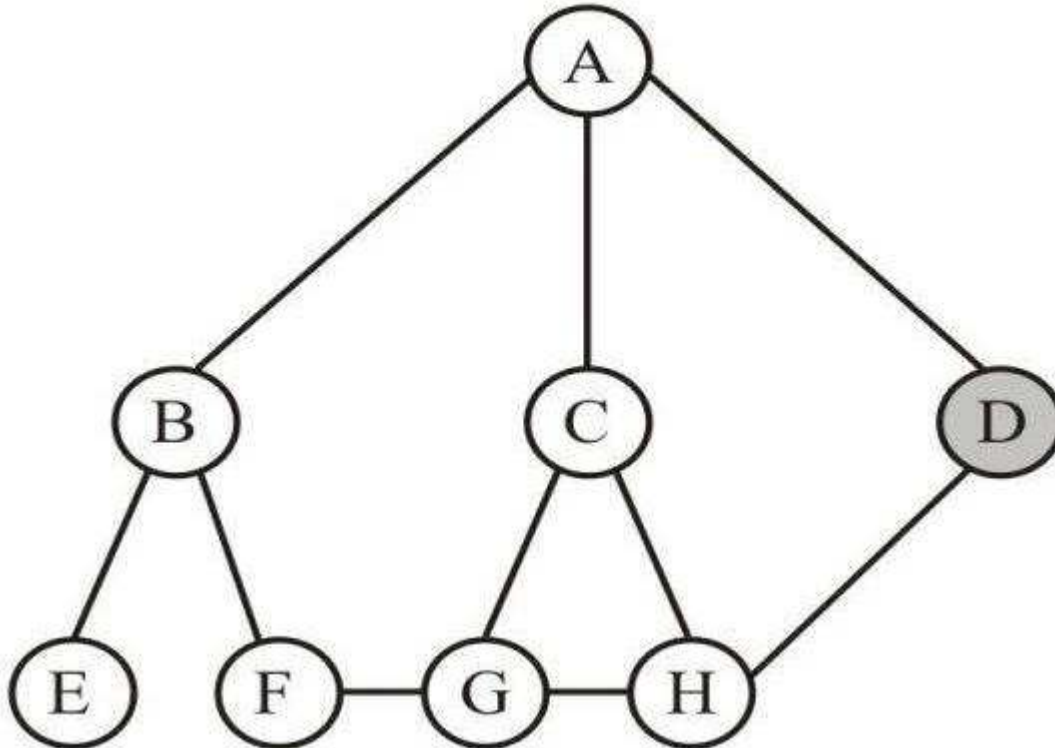
These algorithms typically solve similar pieces of the problem, and then put them together at the end. Mergesort is a classic example of divide and conquer. The main difference between this example and the Fibonacci example is that in a mergesort, the division can (theoretically) be arbitrary, and no matter how you slice it up, you are still merging and sorting. The same amount of work has to be done to mergesort the array, no matter how you divide it up. Solving for $\text{fib}(52)$ requires more steps than solving for $\text{fib}(2)$.

For 100% Result Oriented IGNOU Coaching and Project Training

Call CPD: 011-65164822, 08860352748

Question 5:

(a) For the graph given in Figure below, use DFS to visit various vertices. The vertex B is taken as the starting vertex and, if there are more than one vertices adjacent to a vertex, then the adjacent vertices are visited in lexicographic order.



(b) In context of graph search, explain the minimax principle. Ans-

Minimizing the maximum losses that could occur in an event or situation is this specific type of decision-making principle. Should a strategy fail, this financial and business strategy measures potential results leading to the least amount of regret. By the use of logic, determine and use this strategy. Aids deciding when presented with two various and conflicting strategies.

A principle for decision-making by which, when presented with two various and conflicting strategies, one should, by the use of logic, determine and use the strategy that will minimize the maximum losses that could occur.

This financial and business strategy strives to attain results that will cause the least amount of regret, should the strategy fail.

**For 100% Result Oriented IGNOU Coaching and Project
Training**

Call CPD: 011-65164822, 08860352748

Question 6:

(a) Is there a greedy algorithm for every interesting optimization problems? Justify your Claim.

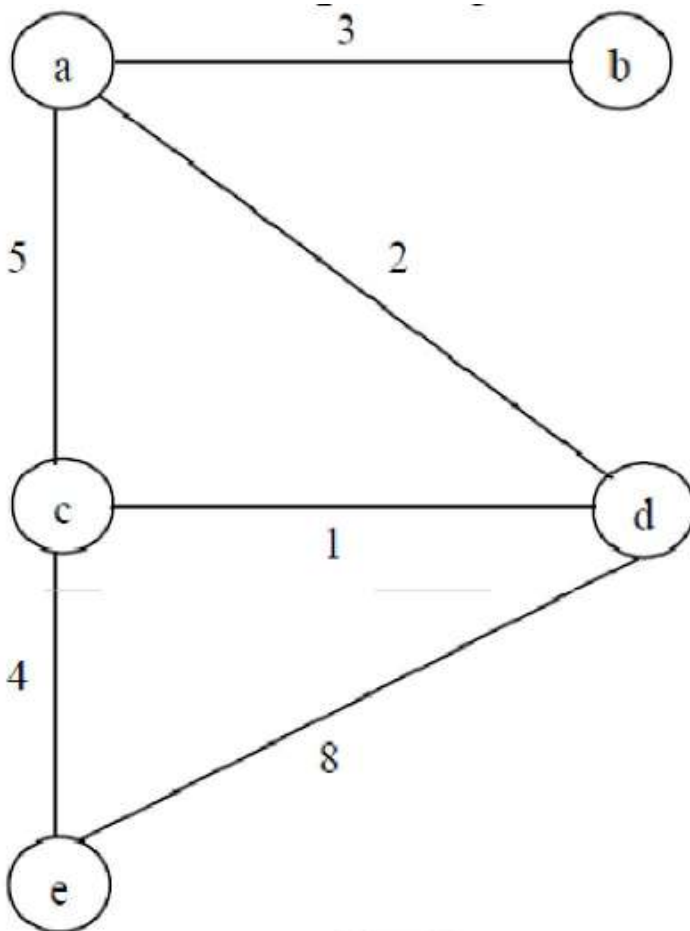
Ans-

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

For example, a greedy strategy for the traveling salesman problem (which is of a high computational complexity) is the following heuristic: "At each stage visit an unvisited city nearest to the current city". This heuristic need not find a best solution but terminates in a reasonable number of steps; finding an optimal solution typically requires unreasonably many steps.

(b) Apply each of (i) Prim's and (ii)Kruskal's algorithms one at a time to find minimal spanning tree for the following graph.

**For 100% Result Oriented IGNOU Coaching and Project
Training
Call CPD: 011-65164822, 08860352748**



Question 7: Write note on each of the following: (20 marks)

- (i) Unsolvability/ undecidability of a problem
- (ii) Halting problem
- (iii) Reduction of a problem for determining decidability
- (iv) Rice theorem

ans-

In computability theory, Rice's theorem states that, for any non-trivial property of partial functions, there is no general and effective method to decide whether

an algorithm computes a partial function with that property. Here, a property of partial functions is called trivial if it holds for all partial computable functions or for none, and an effective decision method is called general if it decides correctly for every algorithm.

- (v) Post correspondence problem
- (vi) NP-complete problem
- (vii) K-colourability problem
- (viii) Independent set problem