Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India

(Autonomous College Affiliated to University of Mumbai)

Make up Examination

Synoptic- May 2018

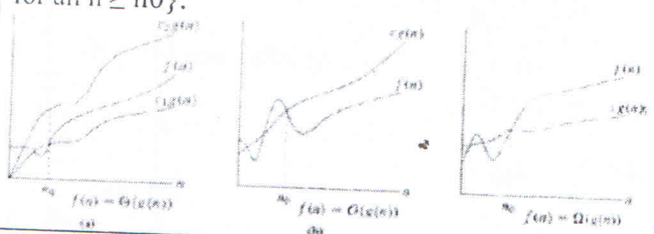
Max. Marks: 100	Duration: 3 Hrs.
Class: S.Y.	Semester: IV
Course Code:MCA43	Branch: M.C.A.
Name of the Course: Design and Analysis of Algorithms	

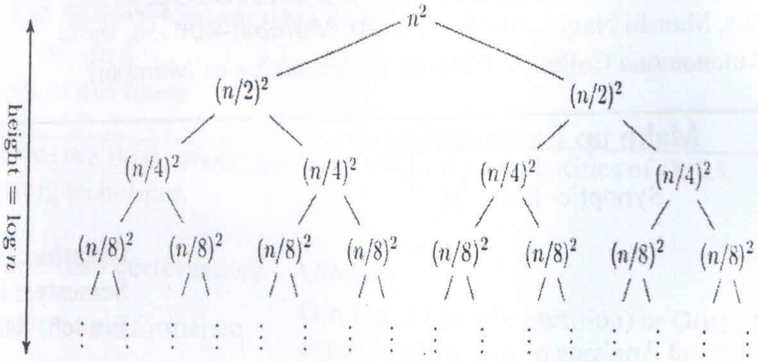
Instruction:

(1) All questions are compulsory

(2) Draw neat diagrams

(3) Assume suitable data if necessary

Q. No.		Max. Marks	CO
Q.1 a)	Explain the various Asymptotic notations with diagrams. O-notation $O(g(n)) = \{f(n)$: there exist positive constants c and n0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n0\}$. Ω-notation $\Omega(g(n)) = \{f(n)$: there exist positive constants c and n0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n0\}$. Θ-notation $\Theta(g(n)) = \{f(n)$: there exist positive constants c1, c2, and n0 such that $0 \leq c1g(n) \leq f(n) \leq c2g(n)$ for all $n \geq n0\}$. 	5	CO1
b)	What is the time complexity of following function fun (). Time Complexity of the above function O(n).	7	CO1

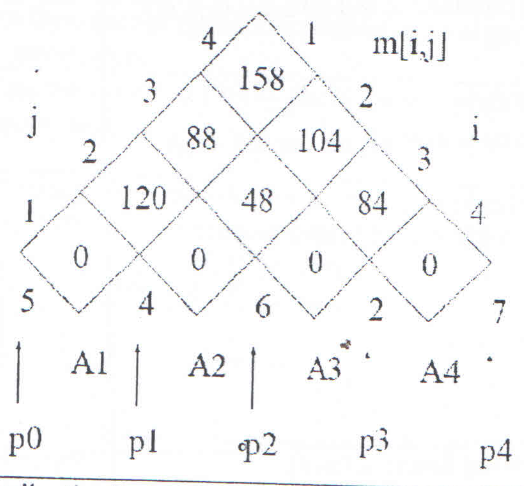
<table>
<tr><td>c)</td><td>Draw the recursion tree for the recurrence T(n) = 2T(n/2) + n2.
up to first 3 levels.


OR

Compare P and NP problems with suitable examples.</td><td>8</td><td>CO1</td></tr>
<tr><td>Q. 2
a)</td><td>Find the LCS of string 1: AGGTAB string2: GXTXAYB

LCS of AGGTAB and GXTXAYB is GTAB</td><td>6</td><td>CO2</td></tr>
<tr><td>b)</td><td>Write Kruskal's algorithm for minimum spanning tree.
OR

Given a chain of four matrices A1,A2,A3,A4 with
p0=5,p1=4,p2=6,p3=2,p4=7.Find m[1,4].
Ans:

m[1,4]=158

</td><td>8</td><td>CO2</td></tr>
<tr><td>c)</td><td>Describe the Dynamic 0/1 Knapsack Problem.
Find an optimal solution for the dynamic programming 0/1 knapsack instance for n=4, m=8, profits are (p1, p2, p3, p4) = (15,10,9,5), weights are (w1,w2,w3,w4)=(1,5,3,4).

The solution is x = (1,0,1,1) i.e. items 1,3, and 4 are selected. value of</td><td>6</td><td>CO2</td></tr>
</table>

	the knapsack is 29.		

Q. 3 a)	Discuss the Graph coloring Problem. What technique is used to solve the problem? Write the algorithm to solve above problem.
	
	Let C[1...j− 1] be a partial coloring for the first j − 1 nodes.
	Color(C,j,k,n)
	if j = n+1 then
	output C
	return or quit
	for i = 1 to k
	C[j] = i
	if valid(C,j,n) then
	Color(C,j+1,k,n)
	where
	Valid(C,j,n)
	for all neighbors v of j with v < j
	if C[v] = C[j] then
	return false
	return true
	
	OR
	Explain the branch and bound strategy and write an algorithm for basic branch and bound strategy.
	BBound (S, U)
	S = {s: s is a feasible state}, U = current upper bound if (FEASIBLE(S) == FALSE) return(¥);
	if (LBOUND(S) >= U) return(¥);
	if (UBOUND(S) < U) U = UBOUND(S); if (LBOUND(S) < U)
	BRANCH(S) -> $S_1, \ldots, S_k$;
	for (i = 0; i < k; i++)
	if (BB(S_i,U) < U) U = BB(S_i);
	return(U);
		7	CO3

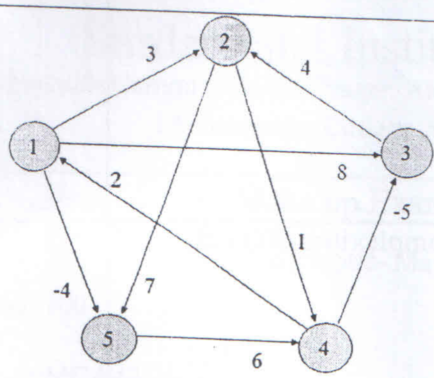
b)	Discuss the Hamiltonian Cycles Problem. What technique is used to solve the problem? Write the algorithm to solve above problem.
	Algorithm HC (Node, String, edges)
	{
	1. If (edges=N and E(SN,Node))
	2. Then
	3. Print (String) //HC found
	4. Exit
	5. If (edges=N)
	6. Then
	7. Return
	8. For each child X of Node and NotIn(X,String)
	9. Do
	10. String = String + info(X)
	11. HC(X,String, edges+1)
	12. Done
	13. Return
	}
		7	CO3

<table>
<tr><td>c)</td><td>Write an algorithm for the subset sum problem.
Algorithm sumofsubset(s,k,r)
{
//generate the left child. note s+w(k)<=M since Bk-1 is true.
X{k]=1;
If (S+W[k]=m) then write(X[1:k]); // there is no recursive call here as W[j]>0,1<=j<=n.
Else if (S+W[k]+W[k+1]<=m) then sum of sub (S+W[k], k+1,r-W[k]);
//generate right child and evaluate Bk.
If ((S+ r- W[k]>=m)and(S+ W[k+1]<=m)) then
{
X{k]=0;
sum of sub (S, k+1, r- W[k]);
}
}

OR

Find the target goal state for the 15-puzzel problem upto 2 levels for the given initial state.
<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td>5</td><td>6</td><td></td><td>11</td></tr>
<tr><td>12</td><td>13</td><td>14</td><td>15</td></tr>
</table></td><td>6</td><td>CO3</td></tr>
<tr><td>Q. 4
a)</td><td>Write an algorithm to find the shortest path using Dijkstra's algorithm.
function Dijkstra(Graph, source):
create vertex set Q

for each vertex v in Graph: dist[v] ← INFINITY
prev[v] ← UNDEFINED
add v to Q
dist[source] ← 0
while Q is not empty:
u ← vertex in Q with min dist[u]
remove u from Q
for each neighbor v of u:
alt ← dist[u] + length(u, v)
if alt < dist[v]: dist[v] ← alt
prev[v] ← u
return dist[], prev[]</td><td>7</td><td>CO4</td></tr>
<tr><td>b)</td><td>Make use of Flyod Warshalls algorithm to find all pair shortest path from every node</td><td>6</td><td>CO4</td></tr>
</table>



Ans:

	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

c) Write an algorithm for Knuth Morris Pratt string matching with example. | 7 | CO4

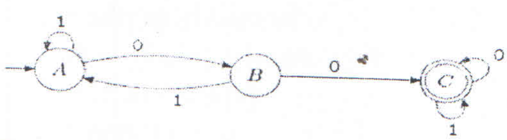
Algorithm kmp_search:
let nP ← 0
while j < length(S) do
 if W[k] = S[j] then

  let j ← j + 1
  let k ← k + 1
  if k = length(W) then
   (occurrence found, if only first occurrence is needed, m may be returned here)
   let P[nP] ← j - k, nP ← nP + 1
   let k ← T[k] (T[length(W)] can't be -1)
 else
  let k ← T[k]
  if k < 0 then
   let j ← j + 1
   let k ← k + 1

OR

Write an algorithm for Rabin Carp string matching algorithm with example.

function RabinKarp(string s[1..n], string pattern[1..m])
 hpattern := hash(pattern[1..m]);
 for i from 1 to n-m+1

	hs := hash(s[i..i+m-1]) if hs = hpattern if s[i..i+m-1] = pattern[1..m] return i return not found		
Q. 5 a)	Derive the Best, Worst and Average time complexities of Quick sorting technique. Worst-case performance O(n2) Best-case performance O(n log n) (simple partition) or O(n) (three-equal keys) Average performance O(n log n) OR Write an algorithm for Vertex cover Problem.APPROXIMATION-VERTEX-COVER(G): C = ∅ E'= G.E while E'≠ ∅: let (u, v) be an arbitrary edge of E' C = C U {u, v} remove from E' every edge incident on either u or v return C	5	CO1
b)	Discuss Dynamic programming and write an algorithm for Optimal binary search tree.	5	CO2
c)	Calculate the time complexity for N-Queens problem. N- Queens : O(N!)	5	CO3
d)	For following deterministic finite automaton obtain 5 –tuple DFA generate 3 strings which are accepted by this DFA  	5	CO4