SRN | | | | | | | | | | | |

PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

UE15CS251

END SEMESTER ASSESSMENT (ESA) B.TECH. 4TH SEMESTER- MAY. 2017

UE15CS251 – DESIGN AND ANALYSIS OF ALGORITHMS [Autonomy, NS]

Time: 3 Hrs | Answer All Questions | Max Marks: 100

Note: All answers must be precise and to the point.

1.	a)	Algo what(m)
		i <- 1
		s <- 0
		while i <= m
		  if m modulo i = 0
		  s <- s + 1
		  i <- i + 1
		return s
		i. What does the algorithm do? State in a sentence.
		ii. What is the number of iterations ?
		iii. Improve the algorithm to decrease the number of iterations.
			4

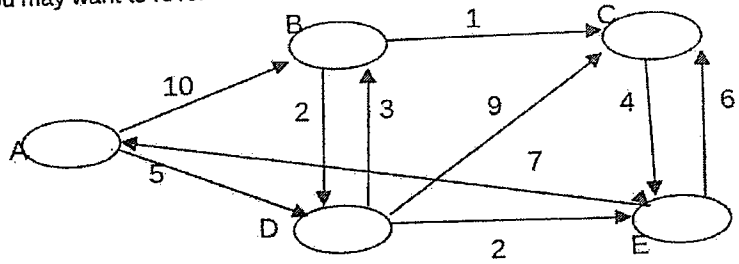
	b)	Develop the recurrence relationship of search in height balanced binary search tree and solve it.	6

	c)	Algorithm Mystery(n)
		for i <- 1 to n do
		  j <- n
		  while j >= 0 do
		  // do some const time operation
		  j <- j / 2
		
		My claim : Complexity is $\Theta(n^2)$.
		Prove or disprove my claim.
			4

	d)	Algorithm GE(a[0..n-1][0 .. n-1]) // A is a 2D array of real numbers
		for i <- 0 to n - 2
		  for j <- i + 1 to n - 1
		  for k <- i to n
		  a[j,k] <- a[j, k] - a[i, k] * a[j, i] / a[i, i]
		1. Find the time efficiency class of this algorithm and express as a summation.
		2. Point out the glaring inefficiency in this algorithm.
			6

2.	a)	This algorithm is supposed to find all matches of the string p in string t.
		Fill up the blanks to satisfy the requirement of this algorithm.
		algo find_all(t[0 .. n - 1], p[0 .. m - 1])
		i <- 0
		while i <= _____
		  j <- _____
		  while j ___ m and p[j] = t[_____]
		  j <- j + 1
		  if j = m
		  disp t[i .. i + m]
		  i <- i + _____
		What should be the difference if the requirement is
		i)To find all overlapping matches
			7

SRN | | | | | | | | | | |

		ii)To avoid overlapping matches **Hint:** aa in aaa : 2 matches if we consider overlapping matches; otherwise 1	
	b)	Develop an algorithm to find the number of connected components of a graph(G). Choose a suitable data structure to represent the graph. **algorithm** ConnectedComponents(G) count <= 0 // TODO return count	6
	c)	Develop an algorithm to seach for a key in a 2-3 tree with the following fields in each node. Leftptr , key1 , midptr ,key2 ,rightptr. Algo search(root, key)	7
3.	a)	Develop an algorithm to find the intersection of two sequences. You may use pre-sorting technique. Assume that a sort function exists.	7
	b)	i. Find the next two permutations using Johnson Trotter algorithm. < < < < >. 3 1 4 2 5 ii. Find the next two permutations using lexicographic order. 3 1 4 2 5	6
	c)	Develop an algorithm which takes as input the output of Floyd's algorithm and converts it into the output of Warshall's algorithm. Algo convert(res_floyd[1 .. n][1 .. n], res_warshall[1 .. n][1 .. n])	7
4.	a)	Apply Dijkstra's algorithm to find **single destination shortest paths** for the given graph with respect to the destination node E . **Hint** : you may want to reverse the directions of edges to change destination to source. 	6
	b)	Apply Horspool's algorithm to search for the pattern AT_THAT in the text i. WHICH_FINALLY_HALTS_ _AT_THAT ii. WHAT_IS_THIS_HAT Count the number of comparisons in each case.	8
	c)	Find the solution for the following instance of the knapsack problem using dynamic programming. Capacity W = 12	6

item	weight	No. of items	Value / item
1	4	2	42
2	3	1	12
3	4	2	40
4	5	1	65

5.	a)	Could these be valid codes using Huffman algorithms? Give your reason in a single sentence. i) frequency of a = 100; frequency of e = 120; code for a : 1001 ; code for e : 10111 ii) frequency of x = 30; frequency of y = 20; code for x : 10101 ; code for y : 101010	6

SRN | | | | | | | | | | | | |

	b)	This is a state while solving the 4-queen problem using backtracking. What is the next state. State how you arrive at the next state. Q - - - - - - Q - Q - - - - - -	4
	c)	In branch and bound technique, how can we make an estimate of the optimizing function in the following cases? knapsack problem traveling salesman problem assignment problem	6
	d)	How to check whether a cycle gets formed while adding an edge in i) Prim's algorithm ii) Kruskal's algorithm	4