

# SPPU-BE-COMP-CONTENT – KSKA Git

Total No. of Questions : 8]

SEAT No. :

PB2243

[6263]-81

[Total No. of Pages :3

**B.E. (Computer Engineering)**

**DESIGN AND ANALYSIS OF ALGORITHMS**

**(2019 Pattern) (Semester- VII) (410241)**

*Time : 2½ Hours ]*

*[Max. Marks : 70*

*Instructions to the candidates:*

- 1) Attempt Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.
- 2) Neat diagrams must be drawn wherever necessary.
- 3) Assume suitable data if necessary.
- 4) Figures to the right indicate maximum marks.

- Q1)** a) Write a control abstraction for greedy method. Comment on the time complexity of this abstraction? [6]
- b) Find an optimal solution for the following knapsack instance using greedy method. Number of objects  $n = 5$ . Capacity of knapsack  $m = 100$ . [8]

Objects	Weight	Profit
01	20	10
02	30	20
03	66	30
04	40	40
05	60	50

- c) Comment on the statement "Problem which does not satisfy the principle of optimality cannot be solved by dynamic programming". [4]

OR

- Q2)** a) Write a control abstraction for dynamic programming strategy. Comment on the time complexity of this abstraction? [8]
- b) Consider 4 matrices A1, A2, A3 and A4. The orders of these matrices are given below: [10]

Matrix	Order
A1	3×5
A2	5×4
A3	4×2
A4	2×4

Find the optimal sequence of chain matrix multiplication of these matrices using dynamic programming approach. Clearly give the final sequence and total number of multiplications involved.

*P.T.O.*

# SPPU-BE-COMP-CONTENT – KSKA Git

**Q3) a)** Assume that a graph with  $n$  vertices is represented by an adjacency matrix  $G$ . Let there be “ $m$ ” number of colours available. Write a recursive backtracking algorithm to colour all the vertices of the graph. What is the time complexity of this algorithm? [8]

**b)** Consider three items along with respective weights and value as [9]

	Weight	Value	Value/Weight
O1	5	6	$6/5 = 1.2$
O2	4	5	$5/4 = 1.25$
O3	3	4	$4/3 = 1.3$

Assume the Knapsack capacity  $m = 7$ . Solve this 0/1 Knapsack problem using LC branch and bound method.

OR

**Q4) a)** Compare backtracking with branch and bound method with respect to: search technique, exploration of state space tree and kind of problems that can be solved. [8]

**b)** Consider set  $A$  of five numbers  $\{5, 10, 15, 20, 25\}$ . We wish to find the subset of  $A$  such that sum of the numbers in this subset is equal to 30. Solve this problem to find the first solution using backtracking approach. Show space tree being created. [9]

**Q5) a)** What are randomized algorithms? Enlist and explain in brief the primary reasons for using randomized algorithms. [8]

**b)** What are approximation algorithms? Based on the approximation ratio, classify the approximation algorithms. [9]

OR

**Q6) a)** Explain the methods of amortized analysis. Give suitable example. [8]

**b)** Suppose you are working on an embedded system for a medical device that monitors patient vital signs. The device continuously collects data from various sensors and needs to process and display this information in real-time. The data includes timestamps, temperature readings, heart rate, and blood pressure measurements. Suggest suitable sorting algorithm for this scenario. Clearly justify your answer with respect to key factors. [9]

# SPPU-BE-COMP-CONTENT – KSKA Git

- Q7) a) Write a Rabin-Karp string matching algorithm. Input to the algorithm be: Original text “t” of length n and pattern text being matched is “p” of length m. What is the expected runtime and worst-case runtime of this algorithm? [10]
- b) Write multi-threaded merge sort algorithm. Briefly discuss how does it differ from conventional merge sort. [8]

OR

- Q8) a) Consider the graph represented by an adjacency matrix: [10]

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	0	1	1	0	0
C	1	0	0	0	0	1	1
D	0	1	0	0	0	0	0
E	0	1	0	0	0	0	0
F	0	0	1	0	0	0	0
G	0	0	1	0	0	0	0

Show stepwise process how the distributed breadth first search algorithm works on the above graph.

- b) What do you understand by spawn and sync keywords used in multithreaded programming? Explain with the help of suitable example. [8]

