Sr. No. 003404

May 2023

B.Tech. - IV SEMESTER

Design & Analysis of Algorithms (PCC-CS-404/PCC-CSH-404)

एल्गोरिदम का डिजाईन और विश्लेषण (पीसीसी-सीएस-404/ पीसीसी-सीएसएच-404)

Time(समय): 3 Hours(3 घंटे) Max. Marks(कुल अंक) :75

Instructions: 1. *It is compulsory to answer all the questions (1.5 marks each) of Part -A in short.*

निर्देश: भाग-क के सभी प्रश्नों (प्रत्येक 1.5 अंक) का उत्तर संक्षेप में देना अनिवार्य है

2. *Answer any four questions from Part -B in detail.*

भाग-ख से किन्हीं चार प्रश्नों के उत्तर विस्तार से दीजिए।

3. *Different sub-parts of a question are to be attempted adjacent to each other.*

एक प्रश्न के विभिन्न उप-भागों को एक दूसरे के निकट करने का प्रयास किया जाना है।

PART–A (भाग - क)

Q1 (a) What is the time complexity of the following code? Justify your answer. (1.5)

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
  for (j = 2; j <= n; j = j * 2) {
    k = k + n / 2;
  }}
```

(क) निम्नलिखित कोड की समय जटिलता क्या है ? आपने जवाब का औचित्य साबित करें। (1.5)

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
  for (j = 2; j <= n; j = j * 2) {
    k = k + n / 2;
  }}
```

(b) Sort the following functions in the decreasing order of their asymptotic (big-O) complexity: $f_1(n) = n^{\sqrt{n}}$, $f_2(n) = 2^n$, $f_3(n) = (1.000001)^n$, $f_4(n) = n^{10} * 2^{(n/2)}$ (1.5)

(ख) निम्नलिखित कार्यों को उनकी एसमटोटिक (बिग-ओ) जटिलता के घटते क्रम में क्रमबद्ध करें: (1.5)

$f_1(n) = n^{\sqrt{n}}$, $f_2(n) = 2^n$, $f_3(n) = (1.000001)^n$, $f_4(n) = n^{10} * 2^{(n/2)}$

(c) Differentiate Greedy algorithm and Dynamic programming. (1.5)

(ग) ग्रीडी एल्गोरिदम और डायनमिक प्रोग्रामिंग में अंतर करें। (1.5)

(d) State Job Sequencing with Deadline Problem. Write down time and space complexity if problem solved by Greedy approach. (1.5)

(घ) जॉब सीक्वेंसिंग विद डेडलाइन समस्या का वर्णन करे । ग्रीडी दृष्टिकोण द्वारा समस्या हल होने पर समय और स्पेस जटिलता लिखें। (1.5)

(e) Define Principle of Optimality with suitable example. (1.5)

(ङ) इष्टतमता के सिद्धांत को उपयुक्त उदाहरण के साथ परिभाषित कीजिए। (1.5)

(f) Draw state space tree of 4-Queens problem. (1.5)

(च) 4-क्वींस प्रॉब्लम का स्टेट स्पेस ट्री बनाइए।

(g) Why does Dijkstra's algorithm fail on negative weights? (1.

(छ) डायजस्ट्रा का एल्गोरिदम नकारात्मक भार पर विफल क्यों होता है? (1.5)

(h) Draw binary search trees for the given set of keys and their corresponding frequencies and find the Optimal among them. keys[ ] = {10, 12, 20}, freq[ ] = {34, 8, 50} (1.5)

(ज) कुंजियों के दिए गए सेट और उनकी संबंधित आवृतियों के लिए बाइनरी सर्च ट्री बनाएं और उनमें से इष्टतम खोजें। कुंजियाँ[ ] = {10, 12, 20}, आवृति[ ] = {34, 8, 50} (1.5)

(i) Explain Least Cost Search function for branch and bound algorithm design technique. (1.5)

(झ) ब्रांच और बाउंड एल्गोरिथम डिजाइन तकनीक के लिए लीस्ट कॉस्ट सर्च फंक्शन की व्याख्या करें। (1.5)

(j) What is the importance of approximation algorithm? (1.5)

(ञ) सन्निकटन एल्गोरिदम का महत्व क्या है? (1.5)

PART –B (भाग - ख)

Q2 (a) Solve the following recurrence relation. (10)
(i) $T(n) = 2T(n/2) + \log n$ with $T(1)=1$
(ii) $T(n) = 2T(\sqrt{n}) + 1$ if $n > 2$ and $T(n) = 2$ if $0 < n <= 2$

(क) निम्नलिखित पुनरावृत्ति संबंध को हल करें । (10)
(i) $T(n) = 2T(n/2) + \log n$ with $T(1)=1$
(ii) $T(n) = 2T(\sqrt{n}) + 1$ if $n > 2$ and $T(n) = 2$ if $0 < n <= 2$

(b) How are time and space trade-offs used to optimize the performance of an algorithm? Provide an example of an algorithm that optimizes time at the expense of space and vice-versa. (5)

(ख) एल्गोरिदम के प्रदर्शन को अनुकूलित करने के लिए समय और स्पेस के बदलाव का उपयोग कैसे किया जाता है? एक एल्गोरिदम का उदाहरण प्रदान करें जो स्पेस की कीमत पर समय का अनुकूलन करता है और इसके विपरीत। (5)

Q3 (a) Give a dynamic-programming solution to the 0-1 knapsack problem that runs in O(n/W) time, where n is the number of items and W is the maximum weight of items that the thief can put in his knapsack. (10)
Consider the weights and values of items listed below. The task is to pick a subset of these items such that their total weight is no more than 5 Kgs and their total value is maximized.

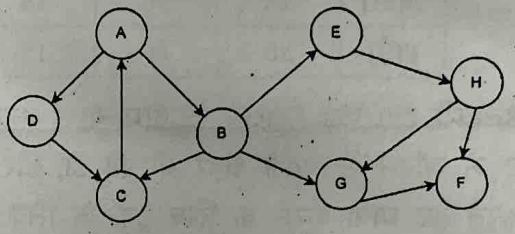
Item No. (आइटम संख्या)	Weight (Kg) वजन (किग्रा)	Values (Rs.) मान (रु.)
	2	3
2	3	4
3	4	5
4	5	6

(क) O(n/W) समय में चलने वाली 0-1 नैपसैक समस्या का डायनेमिक-प्रोग्रामिंग समाधान दें, जहां n वस्तुओं की संख्या है और W उन वस्तुओं का अधिकतम वजन है जो चोर अपने नैपसैक में डाल सकता है। (10)

नीचे सूचीबद्ध आइटम के वजन और मूल्यों पर विचार करें। इन वस्तुओं का एक सबसेट चुंनना है जैसे कि उनका कुल वजन 5 किलोग्राम से अधिक न हो और उनका कुल मूल्य अधिकतम हो।

(b) Consider the given graph. In what order will the nodes be visited using a Breadth First Search and Depth First Search? (Assume starting vertex A) (5)

Note: If a node has multiple neighbors then select which is alphabetically near to node.



(ख) दिए गए ग्राफ पर विचार करें। चौड़ाई पहले खोज और गहराई पहले खोज का उपयोग करके किस क्रम में नोड्स का दौरा किया जाएगा? (मान लें कि शुरुआती वर्टेक्स ए है) (5)

नोट: यदि एक नोड के कई पड़ोसी हैं, तो वर्णानुक्रम में नोड के निकट का चयन करें।

Q4 (a) The N-Queen problem is a classic problem in computer science, where the goal is to place N queens on an N x N chessboard so that no two queens attack each other. (12)

(i) Write a brute-force algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and explain why it is not efficient for large values of N.

(ii) Write a backtracking algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and compare it with the brute-force algorithm.

(क) कंप्यूटर विज्ञान में एन-क्वीन समस्या एक क्लासिक समस्या है, जहां लक्ष्य एन क्वीन्स को एन x एन शतरंज बोर्ड पर रखना है ताकि कोई भी दो क्वीन्स एक-दूसरे पर हमला न करें। (12)

(i) एन-क्वीन समस्या को हल करने के लिए ब्रूट-फोर्स एल्गोरिथम लिखें। अपने

एल्गोरिदम की समय जटिलता का विश्लेषण करें, और समझाएं कि यह एन के बड़े मूल्यों के लिए कुशल क्यों नहीं है।

(ii) एन-क्वीन समस्या को हल करने के लिए बैकट्रैकिंग एल्गोरिथम लिखें। अपने एल्गोरिदम की समय जटिलता का विश्लेषण करें, और इसकी तुलना ब्रूट-फोर्स एल्गोरिदम से करें।

(b) Describe the Traveling Salesman Problem and explain why it is NP-complete. (3)

(ख) ट्रैवलिंग सेल्समैन समस्या का वर्णन करें और समझाएं कि यह एनपी-पूर्ण क्यों है। (3)

Q5 (a) A delivery truck travels between multiple destinations. The truck starts at city A and visits cities B, C, D, and E before returning to A. Design an algorithm to find the shortest path for the truck to travel while minimizing cost. Use following distance matrix to solve the problem. (8)

	A(ए)	B(बी)	C(सी)	D(डी)	E(ई)
A(ए)	0	10	20	15	30
B(बी)	10	0	25	30	20
C(सी)	20	25	0	35	15
D(डी)	15	30	35	0	10
E(ई)	30	20	15	10	0

(क) एक डिलीवरी ट्रक कई गंतव्यों के बीच यात्रा करता है। ट्रक शहर ए से शुरू होता है और ए में लौटने से पहले शहर बी, सी, डी, और ई की यात्रा करता है। लागत को कम करते हुए यात्रा करने के लिए ट्रक के लिए सबसे छोटा रास्ता खोजने के लिए एक एल्गोरिदम डिज़ाइन करें। समस्या को हल करने के लिए निम्नलिखित दूरी मैट्रिक्स का प्रयोग करें। (8)

(b) Write a short note on Randomized algorithms. (7)

(ख) यादृच्छिक एल्गोरिदम पर एक संक्षिप्त टिप्पणी लिखिए। (7)

Q6 (a) Define spanning tree. Write Kruskal's algorithm for finding minimum cost spanning tree. Describe how Kruskal's algorithm is different from Prim's algorithm for finding minimum cost spanning tree. (10)

(क) स्पैनिंग ट्री को परिभाषित कीजिए। न्यूनतम लागत स्पैनिंग ट्री ज्ञात करने के लिए क्रुस्कल एल्गोरिद्म लिखिए। वर्णन करें कि क्रुस्कल का एल्गोरिदम न्यूनतम लागत स्पैनिंग ट्री खोजने के लिए प्रिम के एल्गोरिदम से अलग कैसे है। (10)

(b) What is Topological Sorting? Explain with example. (5)

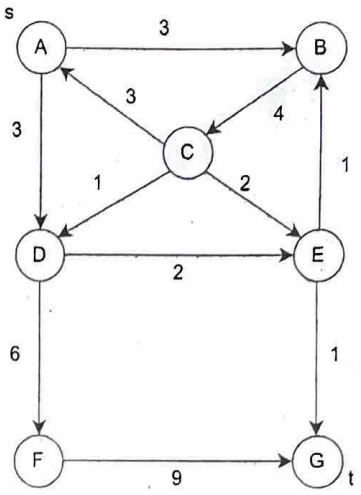
(ख) टोपोलॉजिकल सॉर्टिंग क्या है? उदाहरण सहित समझाइए। (5)

Q7 (a) What is the relationship among P, NP, NP-Hard and NP-Complete problems? Show with the help of a diagram. (8)

(क) पी, एनपी, एनपी-हार्ड और एनपी-पूर्ण समस्याओं के बीच क्या संबंध है? आरेख की सहायता से दर्शाइए। (8)

(b) In a given directed graph with source s and sink t, where each edge in the graph has a non-negative capacity. Find the maximum flow that can be sent from s to t. (7)

(ख) स्रोत s और सिंक t के साथ दिए गए निर्देशित ग्राफ़ में, जहाँ ग्राफ़ के प्रत्येक किनारे की एक गैर-नकारात्मक क्षमता होती है। s से t तक भेजे जा सकने वाले अधिकतम प्रवाह का पता लगाएं। (7)



****

में (1.5)