



PAPER ID-411288

Printed Page: 1 of 4

Subject Code: BCS409

Roll No:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BTECH
(SEM IV) THEORY EXAMINATION 2023-24
DESIGN AND ANALYSIS OF ALGORITHM

TIME: 3 HRS**M.MARKS: 70**

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A**1. Attempt all questions in brief.****2 x 7 = 14**

a.	Explain why Quick Sort is preferred for arrays. Quick Sort को arrays के लिए क्यों प्राथमिकता दी जाती है, समझाएं।
b.	Analyze the best-case time complexities of Heap Sort. Heap Sort के best case समय जटिलता का विश्लेषण करें।
c.	How does merging operation in Binomial Heaps operation differ from the same operation in binary heaps? Binomial Heaps में merging संचालन binary heaps में समान संचालन से कैसे भिन्न है?
d.	Discuss the advantages of using Tries over hash tables for implementing a dictionary. dictionary को लागू करने के लिए Tries के उपयोग के लाभ पर चर्चा करें।
e.	Discuss the application of the Convex Hull problem in computer graphics. कंप्यूटर ग्राफिक्स में Convex Hull समस्या के अनुप्रयोग पर चर्चा करें।
f.	Compare Dynamic Programming in solving the All-Pairs Shortest Paths problem using Floyd's algorithm with the Bellman-Ford algorithm for the same problem. Floyd's एल्गोरिदम का उपयोग करके All-Pairs Shortest Paths समस्या को हल करने में Dynamic Programming और समान समस्या के लिए Bellman-Ford एल्गोरिदम के साथ इस दृष्टिकोण की तुलना करें।
g.	Discuss the practical applications of string matching in search engines. सर्च इंजन में string matching के व्यावहारिक अनुप्रयोगों पर चर्चा करें।

SECTION B**2. Attempt any three of the following:****7 x 3 = 21**

a.	Explain the concept of "Growth of Functions" in the context of algorithm complexity. How do Big O, Big Omega, and Big Theta notations help in this analysis? एल्गोरिदम की जटिलता के संदर्भ में "Functions की Growth" की अवधारणा समझाएं। Big O, Big Omega, और Big Theta notations इस विश्लेषण में कैसे मदद करते हैं?
b.	What is a Skip List? Explain its structure and the algorithms for insertion, deletion, and search operations in detail. Compare Skip Lists with balanced trees like AVL and Red-Black Trees in terms of efficiency and ease of implementation.



PAPER ID-411288

Printed Page: 2 of 4

Subject Code: BCS409

Roll No:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BTECH
(SEM IV) THEORY EXAMINATION 2023-24
DESIGN AND ANALYSIS OF ALGORITHM

TIME: 3 HRS**M.MARKS: 70**

	Skip List क्या है? इसकी संरचना और insertion, deletion, और search संचालन के एल्गोरिदम का विस्तार से वर्णन करें। दक्षता और कार्यान्वयन में सरलता के संदर्भ में Skip Lists की AVL और Red-Black Trees जैसी balanced trees से तुलना करें।
c.	Describe the matrix multiplication algorithm using the Divide and Conquer approach. Provide a detailed implementation and analyze its time complexity. Divide and conquer दृष्टिकोण का उपयोग करके matrix multiplication एल्गोरिदम का वर्णन करें। विस्तृत कार्यान्वयन प्रदान करें और इसकी समय जटिलता का विश्लेषण करें।
d.	Describe the Sum of Subsets problem and its solution using backtracking. Discuss the advantages of using backtracking for solving this problem compared to other approaches. Sum of Subsets समस्या और backtracking का उपयोग करके इसके समाधान का वर्णन करें। अन्य दृष्टिकोणों की तुलना में इस समस्या को हल करने के लिए backtracking के उपयोग के लाभों पर चर्चा करें।
e.	Explain the Fast Fourier Transform (FFT) algorithm and its applications. Provide a detailed step-by-step implementation of the FFT algorithm and analyze its time complexity. Discuss the importance of FFT in signal processing and image compression. Fast Fourier Transform (FFT) एल्गोरिदम और इसके अनुप्रयोगों को समझाएं। FFT एल्गोरिदम का विस्तृत चरण-दर-चरण कार्यान्वयन प्रदान करें और इसकी समय जटिलता का विश्लेषण करें। सिग्नल प्रोसेसिंग और इमेज कम्प्रेसन में FFT के महत्व पर चर्चा करें।

SECTION C**3. Attempt any one part of the following:****7 x 1 = 7**

(a)	Compare the efficiency of sorting algorithms: Bubble Sort, Insertion Sort, and Selection Sort. Provide examples where each is the most appropriate. सॉर्टिंग एल्गोरिदम: Bubble Sort, Insertion Sort, और Selection Sort की दक्षता की तुलना करें। उदाहरण दें जहां प्रत्येक सबसे उपयुक्त हो।
(b)	Explain the concept of Sorting in Linear Time with examples. How do algorithms like Counting Sort and Radix Sort achieve this? उदाहरणों के साथ Linear Time में Sorting की अवधारणा समझाएं। Counting Sort और Radix Sort जैसे एल्गोरिदम इसे कैसे प्राप्त करते हैं?

4. Attempt any one part of the following:**7 x 1 = 7**

(a)	Implement a Binomial Heap and explain its operations such as union, insertion, and deletion in detail. Analyze the time complexity of each operation and discuss the practical applications of Binomial Heaps in computer science.
-----	--



PAPER ID-411288

Roll No:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BTECH
(SEM IV) THEORY EXAMINATION 2023-24
DESIGN AND ANALYSIS OF ALGORITHM

TIME: 3 HRS**M.MARKS: 70**

	Binomial Heap को लागू करें और इसके संचालन जैसे union, insertion, और deletion को विस्तार से समझाएं। प्रत्येक संचालन की समय जटिलता का विश्लेषण करें और कंप्यूटर विज्ञान में Binomial Heaps के व्यावहारिक अनुप्रयोगों पर चर्चा करें।
(b)	Discuss the structure of Fibonacci Heaps and their operations, including insertion, deletion, decrease key, and union. Explain how Fibonacci Heaps improve the efficiency of Dijkstra's algorithm and provide a detailed analysis of their amortized time complexity. Fibonacci Heaps की संरचना और उनके संचालन, जिसमें insertion, deletion, decrease key, और union शामिल हैं, पर चर्चा करें। Fibonacci Heaps Dijkstra's एल्गोरिदम की दक्षता को कैसे सुधारते हैं, समझाएं और उनकी amortized समय जटिलता का विस्तृत विश्लेषण प्रदान करें।

5. Attempt any one part of the following:**7 x 1 = 7**

(a)	Explain the Knapsack problem using the Greedy method. Provide a detailed example where the Greedy approach is applied to solve the Knapsack problem. Discuss why this approach fails for the 0/1 Knapsack problem and compare it with the Dynamic Programming approach. Greedy विधि का उपयोग करके Knapsack समस्या को समझाएं। एक विस्तृत उदाहरण दें जहां Greedy दृष्टिकोण Knapsack समस्या को हल करने के लिए लागू किया गया है। यह दृष्टिकोण 0/1 Knapsack समस्या के लिए क्यों विफल होता है और इसे Dynamic Programming दृष्टिकोण के साथ तुलना करें।
(b)	Describe Prim's algorithm for finding the Minimum Spanning Tree. Provide a step-by-step implementation and analyze its time complexity with a suitable example. Discuss the practical applications of Prim's algorithm in network design and optimization. Minimum Spanning Tree खोजने के लिए Prim's एल्गोरिदम का वर्णन करें। चरण-दर-चरण कार्यान्वयन प्रदान करें और एक उपयुक्त उदाहरण के साथ इसकी समय जटिलता का विश्लेषण करें। नेटवर्क डिजाइन और अनुकूलन में Prim's एल्गोरिदम के व्यावहारिक अनुप्रयोगों पर चर्चा करें।

6. Attempt any one part of the following:**7 x 1 = 7**

(a)	Discuss Floyd's algorithm for the all-pairs shortest path problem. Provide a detailed implementation and analyze its time complexity. Compare this algorithm with Warshall's algorithm and discuss the scenarios where Floyd's algorithm is more efficient. all-pairs shortest path समस्या के लिए Floyd's एल्गोरिदम पर चर्चा करें। विस्तृत कार्यान्वयन प्रदान करें और इसकी समय जटिलता का विश्लेषण करें। इस एल्गोरिदम की Warshall's एल्गोरिदम के साथ तुलना करें और उन परिदृश्यों पर चर्चा करें जहां Floyd's एल्गोरिदम अधिक कुशल है।
(b)	Describe the Hamiltonian Cycle problem and its solution using backtracking. Compare the backtracking approach with other methods for solving the Hamiltonian Cycle problem.



PAPER ID-411288

Printed Page: 4 of 4

Subject Code: BCS409

Roll No:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

BTECH
(SEM IV) THEORY EXAMINATION 2023-24
DESIGN AND ANALYSIS OF ALGORITHM

TIME: 3 HRS**M.MARKS: 70**

	Hamiltonian Cycle समस्या और backtracking का उपयोग करके इसके समाधान का वर्णन करें। Hamiltonian Cycle समस्या को हल करने के लिए अन्य विधियों के साथ backtracking दृष्टिकोण की तुलना करें।
--	--

7. Attempt any one part of the following:**7 x 1 = 7**

(a)	Discuss the theory of NP-Completeness and its implications in computer science. Provide a detailed explanation of the concept of NP-Complete problems and illustrate with examples. NP-Completeness के सिद्धांत और कंप्यूटर विज्ञान में इसके निहितार्थों पर चर्चा करें। NP-Complete समस्याओं की अवधारणा का विस्तृत स्पष्टीकरण प्रदान करें और उदाहरणों के साथ स्पष्ट करें।
(b)	Describe the Randomized Algorithms and their applications. Provide a detailed implementation of the Quick Sort algorithm using randomization and analyze its time complexity. Randomized एल्गोरिदम और उनके अनुप्रयोगों का वर्णन करें। Randomization का उपयोग करके Quick Sort एल्गोरिदम का विस्तृत कार्यान्वयन प्रदान करें और इसकी समय जटिलता का विश्लेषण करें।