PARUL UNIVERSITY - Faculty of Engineering and Technology

Department of Computer Science & Engineering

SYLLABUS FOR 4th Sem BTech PROGRAMME

Competetive Coding Level -2A

Type of Course: BTech

Prerequisite: Computer Programming and Basic Syntaxes

Rationale: Learner will be able to attempt all the coding examinations in the market

:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lect Hrs/	Tut Hrs/	Lab Hrs/		T	P	T	CE	P	
3	0	0	3	60	-	20	20	-	100

Lect - Lecture, **Tut** - Tutorial, **Lab** - Lab, **T** - Theory, **P** - Practical, **CE** - CE, **T** - Theory, **P** - Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Introduction**: **Stacks** -Construction -Operations -Stack stacked Queues -Construction -Operations -Queue Queued	10%	6
2	LinkedLists -Construction -operations -Merging Two Sorted Lists -Merge Point of Two Sorted Lists -nth node from the end -Swap Nodes Pair wise	15%	8
3	Trees -Introduction -Types of Trees -Binary Trees -Tree Traversals -Views of Binary Tree(Top view, Bottom View)	10%	5

4	Binary Trees -Mirrored Trees -Sum Tree or Not -Height and Diameter of a Binary Tree -Sum from Root to leaf Path -Ancestors of a Binary tree -Lowest Common Ancestor of a Binary Tree -Binary Search Tree -Construction -Insertion and Deletion	10%	5
5	Priority Queues -Construction -Max Heap -Min Heap -Heap Sort	10%	4
6	Introduction to Hashing. Index Mapping (or Trivial Hashing) Separate Chaining for Collision Handling. Open Addressing for Collision Handling. Double Hashing. Load Factor and Rehashing.	15%	8
7	Introduction to Tries, Making a Trie Node Insert, Search and Remove operation implementation in Tries, Types of Tries, Huffman Coding	10%	5
8	Longest Word with all Prefixes Number of Distinct Substrings in a String Maximum XOR of Two numbers in an Array Count Words in a Trie	15%	9

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

Introduction to Algorithms By Thomas H . Cormen, Charles E. Leiserson: ...
Competitive Programming 3 by Steven Halim: ...
Guide to Competitive Programming by Antti Laaksonen: ...
Programming Challenges by Steven S Skiena: ...
The Algorithm Design Manual By Steven S Skiena:

Course Outcome:

1. Judge time complexity rules during problem solving.

2.Apply sorting algorithms to data structures to solve problems.

3. Select the best data structure to solve the given problem.

4. Solve given problems using different Problem Solving Techniques.

PARUL UNIVERSITY - Faculty of Engineering and Technology

Department of Computer Science & Engineering

SYLLABUS FOR 4th SEM B. Tech PROGRAMME

Competitive Coding Level-2A

Type of Course: B. Tech

Prerequisite: proficiency in a programming language (e.g., C++, Python) and a strong grasp of data structures and algorithms, with a focus on problem-solving skills and efficient code implementation. Familiarity with common coding platforms (e.g., Codeforces, LeetCode) is also beneficial.

Rationale: Competitive coding sharpens problem-solving skills, enhances algorithmic thinking, and fosters quick and efficient coding practices. It provides a platform for continuous learning, challenges individuals to tackle diverse problems, and fosters a competitive spirit that's valuable in technical interviews and real-world software development.

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lect Hrs/	Tut Hrs/	Lab Hrs/ Week		T	P	T	CE	P	
8	0	8	1	-		-	-		

Lect- Lecture, **Tut** - Tutorial, **Lab** - Lab, **T** - Theory,**P**- Practical, **CE**- CE, **T** - Theory,**P**- Practical

Objectives:

1. Develop strong problem-solving skills, improve algorithmic thinking, and enhance proficiency in coding by tackling a variety of challenging problems.
2. Cultivate the ability to write efficient and optimized code under time constraints, honing the skill of quickly translating algorithmic insights into practical solutions.
3. Gain a competitive advantage in technical interviews and coding assessments, showcasing the ability to tackle diverse coding challenges commonly encountered in job placements and coding competitions.
4. Foster a mindset of continuous learning by regularly engaging with new problems, staying updated on emerging algorithms, and adapting to evolving coding paradigms.

List of Practical's:

1. Write a program for implementing a MINSTACK which should support operations like push, pop, overflow, underflow, display
    a. Construct a stack of N-capacity
    b. Push elements
    c. Pop elements
    d. Top element
    e. Retrieve the min element from the stack

2. Write a program to deal with real-world situations where Stack data structure is widely used
    Evaluation of expression:
    Stacks are used to evaluate expressions, especially in languages that use postfix or prefix notation. Operators and operands are pushed onto the stack, and operations are performed based on the LIFO principle.

3. Write a program for finding NGE NEXT GREATER ELEMENT from an array

4. Write a program to design a circular queue(k) which Should implement the below functions
    a. Enqueue
    b. Dequeue
    c. Front
    d. Rear

5. Write a Program for an infix expression, and convert it to postfix notation. Use a queue to implement the Shunting Yard Algorithm for expression conversion.

6. Write a Program for finding the Product of the three largest Distinct Elements. Use a Priority Queue to efficiently find and remove the largest elements

7. Write a Program to Merge two linked lists(sorted)
8. Write a Program to find the Merge point of two linked lists(sorted)
9. Write a Program to Swap Nodes pairwise
10. Write a Program for Building a Function ISVALID to VALIDATE BST
11. Write a Program to Build BST
12. Write a Program to determine the depth of a given Tree by Implementing MAXDEPTH
13. Write a Program to Understand and implement Tree traversals i.e. Pre-Order Post-Order, In-Order
14. Write a Program to perform Boundary Traversal on BST
15. Write a program for Lowest Common Ancestors
16. Write a Program to verify and validate mirrored trees or not
17. Write a Program for a basic hash function in a programming language of your choice. Demonstrate its usage to store and retrieve key-value pairs.
18. Implement a hash table using separate chaining for collision handling. Perform operations like insertion, deletion, and search on the hash table.
19. Write a Program to Implement Two sums using HASHMAP
20. Write a Program to Implement Search, insert, and Remove in Trie
21. Write a Program to Implement Huffman coding
22. Write a Program to find Distinct substrings in a string
23. Write a Program to find The No of Words in a Trie
24. Write a Program to view a tree from left View

25. Write a Program to Traverse a Tree using Level Order Traversal