INDEX

Syllabus

Theory 2021	Lab 2021
Searching and sorting Selection sort Linear search External merge sort Quick sort Hashing Data Structures Linked list Arrays Stack Tree Graph Theoretical perspective of computing State Machines Complexity Languages and compilers Types of Grammar Notations Generations of PL How compilers are developed Artificial Intelligence Knowledge based systems ANN Computer vision	OOP DB Access Web Automation Binary Files Recursion Intro to SE

LABS

List	**Hot topics**	
Programming questions of ICS-1 QP 2021 Frequency of alphabets in a Text File + Binary File Binary Files Writing and reading it back int/float/char/short/long/double Bit-operators Writing n bits that are available as strings “011011110101011100001” etc. Data compression using Hauffman Algorithm Hamming codes and algorithm for coding and verification Binary Search Tree search algorithm Linked list traversing Infix to Postfix Evaluation of Postfix Graph structure - reachability and shortest path Gauss Jordan method - input pivot element and reduce Inherit GJ for automated solution	**ANN, Learning, Deep Learning** **DSS** **Image processing** **IoT** **Cloud** **Data science** **Linux from scratch** **Mobile Programming** **Web programming** **DB access** **Web Automation** **Office Automation**	

Inherit Inherited-GJ for inverse of a matrix DSS for LP Programs Maze Solver Matrix Operations - -Add, subtract, Multiply two matrices Inherit Matrix Operations for Linear Algebra toolkit - Reachability, Shortest path, Euler path Vector Image Operations - rotate, resizem Move, Shear Image processing - Gray scale, color filters, smoothing/blur, Sharpen Expression Verifier Web automation using Selenium DB access from High Level languages		

BSCS-302 Teaching ideas and planning

Session	Learning Objectives	Main discussion points	Exam Questions	Lab
1	1. Linear/Sequential search 2. Inplace sorting defn 3. Selection Sort 4. Functions			
2	**Theory** 1. Binary Search 2. Introduction to Data Structures 3. Linked list 4. Binary Search Tree 5. Recursion **Lab** 1. Binary Search 2. Linked List traversal 3. Searching a Binary Search Tree 4. Recursion		1. What is root node?/ leaf node/ parent node/ child node? 2. A Node in binary tree can attach to how many nodes? 3. Pre condition for searching a Binary Search Tree? 4. Recursive function to calculate power / factorial 5. Binary Search by recursion 6. Execution trace of recursive program 7. What is recursion?	1. Binary Search program in python 2. Practice of recursive function

3	1. Generic definition of linked list
	2. Pointers and malloc
	3. High level v/s low level languages - Middle level language = C language
	4. Boot process, File Storage
	5. Primary v/s Secondary storage
	6. LAB: Binary Search, Recursion
	
	LAB
	Bit operators
	Binary Files
		Bit Operators
		How (1) positive numbers are stored (2) negative number are stored
		+ve: as binary
		-ve: as two's complement
		Operators: & | ~ ^(No Xor in python) << >>
		Operators: &= |= ^= ((No Xor in python) >>= <<=
		`[ord(character) for character in "€uro"]`
		`[8364, 117, 114, 111]`
		€=10000010101100_2
		u=1110101_2
		etc.
		x << y
		Returns x with the bits shifted to the left by y places (and new bits on the right-hand-side are zeros). This is the same as multiplying x by 2**y.
		x >> y
		Returns x with the bits shifted to the right by y places. This is the same as //'ing x by 2**y.
		x & y
		Does a "bitwise and". Each bit of the output is 1 if the corresponding bit of x AND of y is 1, otherwise it's 0.
			1. What kind of tasks cannot be done with highlevel languages?
			2. Abbrevation (BIOS, ISR, POST)
			3. What is Interrupt Service routine?	

		x | y
		Does a "bitwise or". Each bit of the output is 0 if the corresponding bit of x AND of y is 0, otherwise it's 1.
		
		~ x
		Returns the complement of x - the number you get by switching each 1 for a 0 and each 0 for a 1. This is the same as -x - 1.
		
		x ^ y
		Does a "bitwise exclusive or". Each bit of the output is the same as the corresponding bit in x if that bit in y is 0, and it's the complement of the bit in x if that bit in y is 1.
		
		Notice that bit-length, which is the number of binary digits, varies greatly across the characters. The euro sign (€) requires fourteen bits, while the rest of the characters can comfortably fit on seven bits.
		
		Note: Here's how you can check the bit-length of any integer number in Python:
		
		`>>>`
		`>>> (42).bit_length()`		

		6
		Without a pair of parentheses around the number, it would be treated as a floating-point literal with a decimal point.
		
		`>>> for char in "€uro":`
		`...  print(char, len(char.encode("utf-8")))`
		`...`
		`€ 3`
		`u 1`
		`r 1`
		`o 1`
		
		`def xor(a, b):`
		`  return (a and not b) or (not a and b)`		

4	1. Tree Traversing methods - In-Order, Pre-Order and Post-Order 2. Binary Search Tree v/s Binary Tree v/s Tree 3. Data compression using Huffman Algorithm as Applications of Binary Tree			

	4. Stack data structure - Push, Pop 5. Applications of Stack Data Structure - verify all opening brackets in an expression have also their matching closing brackets, Palindrome, Infix to Postfix Conversion			

----------Session 1: (06-Sep-2021)----------

Learning Objectives

Lecture Summary Points

Exam Questions

Lab Discussion

Lab tasks/Coding questions

----------Session 2: (13-Sep-2021)----------

Learning Objectives

Lecture Summary Points

- Malloc function
- Low level and high level features in C lang language
- Some of the things that we can't do from high level languages like OS development, device drivers etc.
- Interrupts of OS

1. Interrupt service routine(main defination, detail).
2. Interrupt no. 5 and its function.

- Firmware
    1. What is firmware.
    2. Power on Self test.
    3. BIOS interrupt.
    4. C lang can access firmware functions.

- Shadow ROM

- What happens when files are deleted. How deleted files are saved in directory file. We can access to hard disk by C language.

- Difference in primary and secondary storage

Exam Questions

4.

Lab Discussion

- Recursion
    1. Step by Step execution
    2. Factorial program by recursion.
    3. Power program by recursion.

- Binary Search Tree
    1. What is root node
    2. How to make binary tree

3. How to search number from it
4. Leaf nodes/ child nodes/ parent nodes
5. Types of Sub Tree

● Hints to make Binary Search program in python language.

Lab tasks/Coding questions

----------Session 3: (20-Sep-2021)----------

Learning Objectives

Lecture Summary Points

- Binary Search Tree
    1. Diagramatic structure
    2. How to search number in binary search tree
- Tree Traversing methods
    1. In order
    2. Pre Order
    3. Post order
- Difference in Tree / Binary Tree / Binary Search Tree
- Structure of Expression Tree
- Application of Trees
- Hauffman Tree:
    How to make it,

Its advantage,
How to traverse it and set binary for characters,
How it works

- Encoding scheme (definition)
- ASCII encoding scheme (Basic)

Exam Questions

- Insert given numbers into binary search tree and create its tabular structure in same sequence.
- Write a program to find leaf node
- Traversing of tree
- What is Difference in Tree / Binary Tree / Binary Search Tree?
- How is data traverse in In-order traversal? What is unique in it?

Lab Discussion

- How to calculate frequency of characters in a file
- Ord() Function

Lab tasks/Coding questions

- Write a program to calculate frequency of characters in a text file
- Home task:
  Matrix program(Addition, subtraction, multiplication)
  Gauss Jordan method program