6IT1A- COMPUTER NETWORKS (Common to CS & IT)

Class: VI Sem. B.Tech.	Evaluation
Branch: Info. Tech **Engg. Schedule per** **Week Lectures: 3**	**Examination Time = Three (3) Hours** **Maximum Marks = 100** **[Mid-term (20) & End-term (80)]**

Units	Contents of the subject
I	Network layer-design issue, routing algorithms: Distance vector, link state, hierarchical, Broadcast routing. Congestion control: congestion prevention policies, congestion control in Datagram subnets, load shedding, jitter control, Leaky bucket and token bucket algorithms.

NOTE: The first 2 lectures shall be devoted to review of the basis architectures and responsibilities of different layers.

II	Internetworking: Differences in networks, Tunneling, Internetwork routing, Fragmentation Network layer in the Internet: IPv4 classful and classless addressing, subnetting Network layer protocols(only working and purpose; packet headers etc. not included), Differences in IPV6 over IPV4. Routing to Mobile Hosts and Mobile IP
III	Elements of transport protocols: addressing, connection establishment and release, flow control and buffering, multiplexing and demultiplexing, crash recovery, introduction to UDP protocol. Principles of Reliable Data Transfer: Reliable data transfer over a perfectly reliable channel, Channel with bit errors and Lossy Channel with bit errors.
IV	Transport Layer in the Internet: Introduction to TCP, TCP service Model, TCP Header and segment structure, TCP connection establishment and release, transmission policy, timer management, Transactional TCP. Mobile TCP TCP Congestion Control: Fairness, TCP delay modeling.
V	Application Layer: World Wide Web (WWW), Domain Name System (DNS), E-mail, File Transfer Protocol (FTP), Introduction to Network security. P2P File Sharing: Centralized Directory, Query flooding, exploiting heterogeneity.

Text/References:

1. Tanenbaum; Computer Network, 4th Ed., Pearson.
2. Kurose; Computer Networking, 3rd Ed., Pearson.
3. Peterson, Davie; Computer Networks, 4rd Ed., ELSEVIER

6IT2A- DESIGN AND ANALYSIS OF ALGORITHMS (Common to CS & IT)

Class: VI Sem. B.Tech.	Evaluation
Branch: Info. Tech Engg. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	BACKGROUND: Review of Algorithm Complexity, Order Notations: definitions and calculating complexity. DIVIDE AND CONQUER METHOD: Binary Search, Merge Sort, Quick sort and strassen's matrix multiplication algorithms. GREEDY METHOD: Knapsack Problem, Job Sequencing, Optimal Merge Patterns and Minimal Spanning Trees.
II	DYNAMIC PROGRAMMING: Matrix Chain Multiplication. Longest Common Subsequence and 0/1 Knapsack Problem. BRANCH AND BOUND: Traveling Salesman Problem and Lower Bound Theory. Backtracking Algorithms and queens problem.
III	PATTERN MATCHING ALGORITHMS: Naïve and Rabin Karp string matching algorithms, KMP Matcher and Boyer Moore Algorithms. ASSIGNMENT PROBLEMS: Formulation of Assignment and Quadratic Assignment Problem.
IV	RANDOMIZED ALGORITHMS. Las Vegas algorithms, Monte Carlo algorithms, randomized algorithm for Min-Cut, randomized algorithm for 2-SAT. Problem definition of Multicommodity flow, Flow shop scheduling and Network capacity assignment problems.
V	PROBLEM CLASSES NP, NP-HARD AND NP-COMPLETE: Definitions of P, NP-Hard and NP-Complete Problems. Decision Problems. Cook's Theorem. Proving NP-Complete Problems - Satisfiability problem and Vertex Cover Problem. Approximation Algorithms for Vertex Cover and Set Cover Problem.

References:

1. Cormen, Leiserson, Rivest: Introduction to Algorithms, Prentice Hall of India.
2. Horowitz and Sahani: Fundamental of Computer algorithms.
3. Aho A.V , J.D Ulman: Design and analysis of Algorithms, AddisonWesley
4. Brassard : Fundamental of Algorithmics, PHI.

6IT3A- THEORY OF COMPUTATION (Common to CS & IT)

Class: VI Sem. B.Tech.	**Evaluation**
Branch: I.T. **Schedule per Week** **Lectures: 3, Tutorial:1**	**Examination Time = Three (3) Hours** **Maximum Marks = 100** **[Mid-term (20) & End-term (80)]**

Units	**Contents of the subject**
I	Finite Automata & Regular Expression: Basic Concepts of finite state system, Deterministic and non-deterministic finite automation and designing regular expressions, relationship between regular expression & Finite automata minimization of finite automation mealy & Moore Machines.
II	Regular Sets of Regular Grammars**:** Basic Definition of Formal Language and Grammars. Regular Sets and Regular Grammars, closure proportion of regular sets, Pumping lemma for regular sets, decision Algorithms for regular sets, Myhell_Nerod Theory & Organization of Finite Automata.
III	Context Free Languages& Pushdown Automata: Context Free Grammars – Derivations and Languages – Relationship between derivation and derivation trees – ambiguity – simplification of CEG – Greiback Normal form – Chomsky normal forms – Problems related to CNF and GNF Pushdown Automata: Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Pushdown automata and CFL - pumping lemma for CFL - Applications of pumping Lemma.
IV	Turing Machines: Turing machines – Computable Languages and functions – Turing Machine constructions – Storage in finite control – multiple tracks – checking of symbols – subroutines – two way infinite tape. Undecidability: Properties of recursive and Recursively enumerable languages – Universal Turing Machines as an undecidable problem – Universal Languages – Rice's Theorems.
V	Linear bounded Automata Context Sensitive Language: Chomsky Hierarchy of Languages and automata, Basic Definition & descriptions of Theory & Organization of Linear bounded Automata Properties of context-sensitive languages

References

1. Aho, Hopcropt and Ullman, Introduction to Automata Theory, Formal Languages and Computation, Narosa
2. Cohen, Introduction to Computer Theory, Addison Wesley.
3. Papadimitriou, Introduction to Theory of Computing, Prentice Hall.

6IT4A- PROGRAMMING IN JAVA

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. **Schedule per Week** **Lectures: 3**	**Examination Time = Three (3) Hours** **Maximum Marks = 100** **[Mid-term (20) & End-term (80)]**

Units	Contents of the subject
I	JAVA: Introduction to Object Orientated Programming, Abstraction, Object Oriented Programming Principles, Features of JAVA, Introduction to Java byte code, Java Virtual machine. PROGRAM ELEMENTS: Primitive data types, variables, assignment, arithmetic, short circuit logical operators, Arithmetic operators, bit wise operators, relational operators, Boolean logic operators, the assignment operators, operator precedence, Decision and control statements, arrays.
II	CONTROL STATEMENTS: Java's Selection Statements, if statement, switch statement, Iteration Statements, while, do-while, for, for-each, Nested Loops, Jump Statements, Using break, Using continue, return. OBJECTS AND CLASSES: Objects, constructors, returning and passing objects as parameter, Nested and inner classes, Single and Multilevel Inheritance, Extended classes, Access Control, usage of super, Overloading and overriding methods, Abstract classes, Using final with inheritance.
III	PACKAGE AND INTERFACES: Defining package, concept of CLASSPATH, access modifiers, importing package, Defining and implementing interfaces. STRING HANDLING: String constructors, special string operations, character extraction, searching and comparing strings, string Buffer class.
IV	EXCEPTION HANDLING: Exception handling fundamentals, Exception types, uncaught exceptions, try, catch and multiple catch statements. Usage of throw, throws and finally FILE HANDLING: I/O streams, File I/O.
V	CONCURRENCY: Processes and Threads, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Joins, Synchronization. APPLET: Applet Fundamentals, using paint method and drawing polygons.

References

1. Herbert Schildt: JAVA 2 - The Complete Reference, TMH, Delhi
2. U.K. Chakraborty and D.G. Dastidar: Software and Systems – An Introduction, Wheeler Publishing, Delhi.
3. Joseph O'Neil and Herb Schildt: Teach Yourself JAVA, TMH, Delhi.

6IT5A- INFORMATION THEORY & CODING

Units	Contents of the subject
I	Introduction to information theory. Uncertainty, Information and Entropy, Information measures for continuous random variables, source coding theorem. Discrete Memory less channels, Mutual information, Conditional entropy.
II	Source coding schemes for data compaction: Prefix code, Huffman code, Shanon-Fane code & Hempel-Ziv coding channel capacity. Channel coding theorem. Shannon limit.
III	Linear Block Code: Introduction to error connecting codes, coding & decoding of linear block code, minimum distance consideration, conversion of non systematic form of matrices into systematic form.
IV	Cyclic Code: Code Algebra, Basic properties of Galois fields (GF) polynomial operations over Galois fields, generating cyclic code by generating polynomial, parity check polynomial. Encoder & decoder for cyclic codes.
V	Convolutional Code: Convolutional encoders of different rates. Code Tree, Trllis and state diagram. Maximum likelihood decoding of convolutional code: The viterbi Algorithm fee distance of a convolutional code.

References

1. Digital Communication, Simon Haykin,

6IT6.1A- ADVANCE TOPICS IN OPERATING SYSTEMS (Common to CS & IT)

Class: VI Sem. B.Tech.	**Evaluation**
Branch: I.T. **Schedule per Week** **Lectures: 3**	**Examination Time = Three (3) Hours** **Maximum Marks = 100** **[Mid-term (20) & End-term (80)]**

Units	Contents of the subject
I	**Operating system structures** – policies & mechanism, Structures- monolithic, layered, virtual machines, micro kernel, exokernels, client- server model. Examples from Linux & Windows. **Threads Advance Concepts**– Libraries- Pthreads, win32 threads, Java threads, Introduction to threading issues, system calls, cancellation, signal handling, thread pool, thread specific data, window threads, Linux threads, Solaris Threads. **Massage Passing System** – Need of Message Passing Systems, design issues, naming, synchronization, Implementation–buffering and delivery; mailboxes; RPC & RMI. Examples Systems – Linux, Windows.
II	**File System-** file system layouts, file system implementation, contagious allocation, link list allocation, indexed allocation, file allocation table, virtual file system, directory implementation- linear list and hash table. File System reliability and integrity. **I/O system**: device drivers/ controllers, busses and interfaces- USB, IDE, SCSI, IEEE1394, RAID system, disk caching and buffering, disk management-disk formatting, RAID Structure, boot block, bad block, swap-space management. **System Security:** Security Problems, Program Threats, System Network Threats, Cryptography as a Security Tool, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Network, Computer Security Classifications. Overview of security in Windows. [4]
III	**The Linux OS:** Unix Vs Linux, Design Principles, Kernel Structure, components Kernel Modules, Shell- usage, types; An overview of- Process Management, Thread Management and Scheduling, Memory Management, Process Scheduling in Linux, File System structure & implementation, I/O Management, Network File System, Inter-process Communications, Booting and login process, security.[3]
IV	**The Window OS:** Design Principles, System Components- Hardware Abstraction layer, Kernel, Executives; Environmental Subsystems- MS-DOS Environment, 16-bit Windows Environment, Win32 API, POSIX subsystem; Exception and Interrupts; An overview of-memory management, process management and thread; Process Scheduling in Windows; File Systems: Internal Layout, recovery, Volume Management and Fault Tolerance, FAT and NTFS, Security features, window registry, OS organizations.[3]

V	**Multiprocessor Operating Systems:** Architecture of Multiprocessor Systems, Overview of Multiprocessor OS, Kernal Structure and Multiprocessing support in Linux & Windows, Process Synchronization- Queued Lock, Spin Lock, Sleep Lock; Process Scheduling. **Multimedia Operating System-** Introduction to Multimedia & Data Compression-concepts, common graphics file formats, common audio file formats; Video server, Process management- real time scheduling; Multimedia file systems, Multimedia file storage mechanisms, Video sever organization.[2] **Mobile Operating System**- Windows CE, Palm OS, Symbian OS, JAVA card, Multos.

Text/Reference Books:

1. DM Dhamdhere: Operating Systems – A Concepts Based Approach, Tata McGraw Hill
2. Achyut S Godbole: Operating Systems, Tata McGraw Hill
3. Tanenbaum: Modern Operating System, Prentice Hall
4. A. Silberschatz and Peter B Galvin: Operating System Principals, Wiley India Pvt. Ltd.
5. Charles Crowly: Operating System A Design – Oriented Approach, Tata McGraw Hill.
6. Bach, Design of Unix Operating Systems.

6IT6.2A- BIO INFORMATICS

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. Schedule per Week Lectures: 3	Examination Time = Three (3) Hours Maximum Marks = 100 [Mid-term (20) & End-term (80)]

Units	Contents of the subject
I	Principles of mass and energy conservation. Thermodynamic properties of pure substances. Equations of state. Correlations for physical and transport properties. Material and energy balances for steady state processes involving single and multiphase systems. Reactive and non-reactive processes.
II	Energy flow in biological systems. Energetic of metabolic path ways. Coupled reactions, microbial growth kinetics, Stoichiometry and energetic analysis of cell growth and product formation. Yield and maintenance coefficients. Oxygen consumption and heat evolution in aerobic cultures. Thermodynamic efficiency of growth.
III	Introduction to fermentation, Design of a an industrial fermented, Process calculations for design of typical industrial fermentation processes. Medium formulation. Batch and continuous heat sterilisation of liquid media. Requirements for process utilities (compressed air, cooling water, steam etc.). Material and energy balances for downstream processing and waste water treatment processes, Bioremediation.
IV	Introduction to industrial bio-process: A historical overview of industrial fermentation processes and products. Role of a bio-process engineer in the biotechnology industry. Outline of the various unit operations involved in an integrated bio-process. Process flow sheeting. A brief survey of organisms, processes products and market economics relating to modern industrial bio-technology. Raw materials for fermentation process: Isolation, preservation and improvement of industrial micro-organisms for overproduction of primary and secondary metabolites. Medium requirements for fermentation process carbon, nitrogen, minerals, vitamins and other nutrients. Examples of simple and complex media. Production of primary metabolites: A brief outline of processes for the production of some commercially important organic acids (e.g. citric acid, itaconic acid, lactic acid, acetic acid, gluconic acid etc.), amino acids (glutamic acid, lysine, aspartic acid, phenylalanine etc.) and alcohols (ethanol 2.3, butanediol etc.)
V	Production of secondary metabolites: Study of production processes for various classes of low molecular weight secondary metabolites. Antibiotics-beta-lactams (penicillins, cephalosporins etc.), aminoglycosides (streptomycin, kanamycin etc.), macrolides (erythromycin), quinines, aromatics etc. Vitamins and steroids. Production of commercially important enzymes and recombinant proteins: Proteases, amylases, lipases, cellulases, pectinases, isomerases and other commercially important enzymes for the food and pharmaceutical industries. Production of recombinant poteins having therapeutic and diagnostic applications. Production of vaccines.

References

1. Bryan Bergerson, Bioinformatics Computing, Pearson Education.
2. Pierre Baldi, Bioinformatics: The Machine Learning Approach, Second Edition (Adaptive Computation and Machine Learning), MIT Press
3. David W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory
4. Warren J. Ewens & Gregory R. Grant, Statistical Methods in Bioinformatics, Springer Verlag
5. Andreas D. Baxevanis & B. F. Francis Ouellette, Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, Wiley Interscience

6IT6.3A- HUMAN COMPUTER INTERFACE (Common to CS & IT)

Class: VI Sem. B.Tech.	**Evaluation**
Branch: I.T. **Schedule per Week** **Lectures: 3**	**Examination Time = Three (3) Hours** **Maximum Marks = 100** **[Mid-term (20) & End-term (80)]**

Units	**Contents of the subject**
I	The Human: input-output channels, Human memory, thinking, emotions, individual differences, psychology and the design of interactive systems. The Computer: Text entry devices with focus on the design of key boards, positioning, pointing and drawing, display devices. The Interaction: Models of interaction, ergonomics, interaction styles, elements of WIMP interfaces, interactivity, experience, engagement and fun. Paradigms for Interaction.
II	Design Process: The process of design, user focus, scenarios, navigation design screen design and layout, iteration & prototyping. Usability Engineering Design rules: Principles to support usability, standards, guidelines, rules and heuristics, HCI patterns.
III	Evaluation Techniques: Definition and goals of evaluation, evaluation through expert analysis and user participation, choosing an evaluation method. User support, requirement, approaches, adaptive help systems, designing user support systems
IV	Cognitive methods: Goals and task hierarchies, linguistic models, challenges of display based systems, physical and device models, cognitive architectures.
V	Communications and collaborations models: Face to Face communication, conversations, Text based communication, group working. Task Analysis: Differences between task analysis and other techniques, task decomposition, knowledge based analysis, ER based analysis, sources of information and data collection, use of task analysis.

References:

1. Human Computer Interaction; Alan Dix et.al, 3rd ed., Pearson

6IT7A- JAVA PROGRAMMING LAB (Common to CS & IT)

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. Schedule per Week Practical Hrs.: 3	Examination Time = Five (4) Hours Maximum Marks = 75 [Sessional/Mid-term (60) & End-term (40)]

Objectives: At the end of the semester, the students should have clearly understood and implemented the following:

1. **Develop an in depth understanding of programming in Java**: data types, variables, operators, operator precedence, Decision and control statements, arrays, switch statement, Iteration Statements, Jump Statements, Using break, Using continue, return.
2. **Write Object Oriented programs in Java**: Objects, Classes constructors, returning and passing objects as parameter, Inheritance, Access Control, Using super, final with inheritance Overloading and overriding methods, Abstract classes, Extended classes.
3. **Develop understanding to developing packages & Interfaces in Java:** Package, concept of CLASSPATH, access modifiers, importing package, Defining and implementing interfaces.
4. **Develop understanding to developing Strings and exception handling:** String constructors, special string operations, character extraction, searching and comparing strings, string Buffer class. Exception handling fundamentals, Exception types, uncaught exceptions, try, catch and multiple catch statements. Usage of throw, throws and finally.
5. **Develop applications involving file handling**: I/O streams, File I/O.
6. **Develop applications involving concurrency**: Processes and Threads, Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Joins, and Synchronization.
7. **Develop applications involving Applet:** Applet Fundamentals, using paint method and drawing polygons.

It is expected that each laboratory assignments to given to the students with an aim to In order to achieve the above objectives

Indicative List of exercises:

1. Programs to demonstrate basic concepts e.g. operators, classes, constructors, control & iteration statements, recursion etc. such as complex arithmetic, matrix arithmetic, tower of Hanoi problem etc.
2. Development of programs/projects to demonstrate concepts like inheritance, exception handling, packages, interfaces etc. such as application for electricity department, library management, ticket reservation system, payroll system etc.
3. Development of a project to demonstrate various file handling concepts.
4. Development of a project to demonstrate various applet concepts.

6IT8A- GUI DESIGN LAB

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. Schedule per Week Practical Hrs : 3	Examination Time = Four (4) Hours Maximum Marks =75 [Sessional/Mid-term (60) & End-term (40)]

S. No.	List of Experiments
1.	Adding buttons, edit fields, and other child-window components
2.	Implement the CObject debugging ability and Common MFC problems
3.	Implement GDI Functions, and the CDC class (Text, Drawing shapes, Bitmaps)
4.	Implementing View class functions I. Interacting with the user II. Event Handling III. Responding to events from different control types
5.	Implementing View class functions I. GDI Functions, and the CDC class II. Text III. Drawing shapes IV. Bitmaps
6.	Implementing Dialog Block class Creating a Dialog box Invoking and displaying Setting and retrieving values from a dialog box
7.	Implementing Dialog Boxes, Completion Database Classes I. ODBC vs. DAO II. Databases and Record sets III. Queries (filtering and ordering)
8.	Printing and Print Preview I. Database-style reports II Common Dialog interface

6IT9A- UML LAB

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. Schedule per Week Practical Hrs : 2	Examination Time = Four (4) Hours Maximum Marks = 50 [Sessional/Mid-term (30) & End-term (20)]

Objectives:

1. The students shall be able to use following modules of UML for system description, implementation and finally for product development.

    - Capture a business process model.
    - The User Interaction or Use Case Model - describes the boundary and interaction between the system and users. Corresponds in some respects to a requirements model.
    - The Interaction or Communication Model - describes how objects in the system will interact with each other to get work done.
    - The State or Dynamic Model - State charts describe the states or conditions that classes assume over time. Activity graphs describe the workflows the system will implement.
    - The Logical or Class Model - describes the classes and objects that will make up the system.
    - The Physical Component Model - describes the software (and sometimes hardware components) that make up the system.
    - The Physical Deployment Model - describes the physical architecture and the deployment of components on that hardware architecture.

The students are expected to use the UML models, prepare necessary documents using UML and implement a system. Some hardware products like digital clock, digital camera, washing machine controller, air conditioner controller, an elctronic fan regulator, an elementary mobile phone etc. may also be chosen.

The students shall be assigned one problem on software based systems and another involving software as well as hardware

6IT10A- DESIGN AND ANALYSIS OF ALGORITHMS Lab (Common to CS & IT)

Class: VI Sem. B.Tech.	Evaluation
Branch: I.T. **Schedule per Week** **Practical Hrs : 3**	**Examination Time = Four (4) Hours** **Maximum Marks = 100** **[Sessional/Mid-term (60) & End-term (40)]**

Objectives: Upon successful completion of this course, students should be able to:

- Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains;
- Apply the algorithms and design techniques to solve problems;
- Analyze the complexities of various problems in different domains.

Suggested Tools: For implementation and estimation of running time on various sizes of input(s) or output(s) as the case may be, Linux platform is suggested.

Suggested Exercises:

A. It is expected that teachers will assign algorithms to the students for estimation of time & space complexity. Algorithms reported in various research journals may be chosen by the teachers.

B. Problem on designing algorithms to meet complexity constraints may be assigned. For example, a problem on design, analysis and implementation for transposing a sparse matrix requiring not more than one pass from the original matrix may be assigned.

C. **A guide to such problems is given below:**

1. **Exploring a Binary Heap:** Consider a binary heap containing n numbers (the root stores the greatest number). You are given a positive integer $k < n$ and a number x. You have to determine whether the k^{th} largest element of the heap is greater than x or not. Your algorithm must take $O(k)$ time. You may use $O(k)$ extra storage.

2. **Merging two search trees**: You are given two height balanced binary search trees T and T', storing m and n elements respectively. Every element of tree T is smaller than every element of tree T'. Every node u also stores height of the subtree rooted at it. Using this extra information how can you merge the two trees in time $O(\log m + \log n)$ (preserving both the height balance and the order)?

3. **Complete binary tree as an efficient data-structure:**
   You are given an array of size n (n being a power of two). All the entries of the array are initialized to zero. You have to perform a sequence of the following online operations :
   1. **(i)** *Add(i,x)* which adds x to the entry A[i].

2. **(ii)** Report *sum(i,j)* = sum of the entries in the array from indices *i* to *j* for any $0 < i < j <= n$.

It can be seen easily that we can perform the first operation in $O(1)$ time whereas the second operation may cost $O(n)$ in worst case. Your objective is to perform these operations efficiently. Give a data-structure which will guarantee $O(\log n)$ time per operation.

4. **Problems on Amortized Analysis**
   a. Delete-min in constant time !!! Consider a binary heap of size n , the root storing the smallest element. We know that the cost of insertion of an element in the heap is $O(\log n)$ and the cost of deleting the smallest element is also $O(\log n)$. Suggest a valid potential function so that the amortized cost of insertion is $O(\log n)$ whereas amortized cost of deleting the smallest element is $O(1)$.
   b. Implementing a queue by two stack
   c. Show how to implement a queue with two ordinary stacks so that the amortized cost of each Enqueue and each Dequeue operation is O(1).

5. **Computing a spanning tree having smallest value of largest edge weight:** Describe an efficient algorithm that, given an undirected graph *G*, determines a spanning tree of *G* whose largest edge weight is minimum over all spanning trees of *G*.

6. **Shortest Path Problems:**
   i. From a subset of vertices to another subset of vertices
   a. Given a directed graph G(V,E), where edges have nonnegative weights. S and D are two disjoint subsets of the set of vertices. Give an O(|V| log |V| + |E|) time algorithm to find the shortest path among the set of paths possible from any node in S to any node in D.

   ii. Paths in Directed Acyclic Graph
   a. Counting the number of paths
   Given two nodes *u,v* in a directed acyclic graph *G(V,E)*. Give an *O(|E|)* time algorithm to count all the paths from *u* to *v*.

   b. **Path passing through a subset of nodes**
   Given two nodes *u,v* and a set of vertices $w_1, w_2,...,w_k$ in a directed acyclic graph *G(V,E)*. Give an *O(|E|)* time algorithm to output a path(if exists) from *u* to *v* which passes through each of the nodes $w_1,...,w_k$. If there is no such path then your algorithm must report that *"no such path exists"*.

7. **Searching for a friend:**
   You are standing at a crossing from where there emerge four roads extending to infinity. Your friend is somewhere on one of the four roads. <u>You do not know on which road he is and how far he is from you</u>. You have to walk to your friend and the total distance traveled by you must be at most a constant times the actual distance of your friend from you. In terminology of algorithms, you should traverse *O(d)* distance, where *d* is the distance of your friend from you.

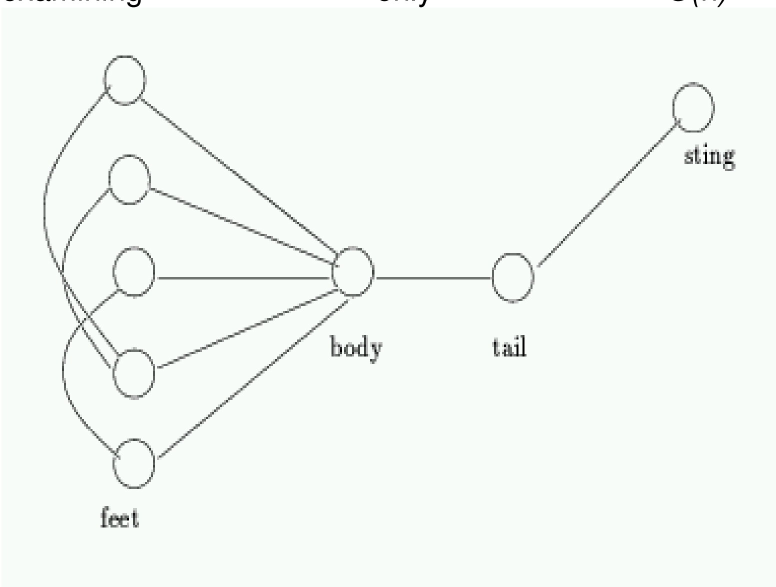
8. **A simple problem on sorted array:** Design an *O(n)*-time algorithm that, given a real number *x* and a sorted array *S* of *n* numbers, determines whether or not there exist two elements in *S* whose sum is exactly *x* .

9. **Finding the decimal dominant in linear time**: You are given *n* real numbers in an array. A number in the array is called a decimal dominant if it occurs more than *n/10* times in the array. Give an *O(n)* time algorithm to determine if the given array has a decimal dominant.

10. **Finding the first one**: You are given an array of infinite length containing zeros followed by ones. How fast can you locate the first one in the array?

11. **Searching for the Celebrity:** Celebrity is a person whom everybody knows but he knows nobody. You have gone to a party. There are total *n* persons in the party. Your job is to find the celebrity in the party. You can ask questions of the form Does Mr. *X* know Mr. *Y* ?. You will get a binary answer for each such question asked. Find the celebrity by asking only *O(n)* questions.

12. **Checking the Scorpion**: An *n*-vertex graph is a *scorpion* if it has a vertex of degree 1(the sting) connected to a vertex of degree two (the tail) connected to a vertex of degree *n-2* (the body) connected to the other *n-3* (the feet). Some of the feet may be connected to other feet. Design an algorithm that decides whether a given adjacency matrix represents a scorpion by examining only *O(n)* entries.



13. **Endless list:** You are having a pointer to the head of singly linked list. The list either terminates at null pointer or it loops back to some previous location(not necessarily to the head of the list). You have to determine whether the list loops back or ends at a null location in time proportional to the length of the list. You can use at most a constant amount of extra storage.

14. **Nearest Common Ancestor**:
Given a rooted tree of size *n*. You receive a series of online queries: "Give nearest common ancestor of *u, v* ". Your objective is to preprocess the tree in *O(n)* time to get a data structure of size *O(n)* so that you can answer any such query in *O(log n)* time.

6IT11A- Humanities and Social Sciences (Common to CS & IT)

Class: VI Sem. B.Tech.	Evaluation
Branch: Information Technology Schedule per Week Practical: 2	Examination Time = Three (3) Hours Maximum Marks = 50 [Sessional/Mid-term (30) & End-term (20)]

Units	Contents of the subject
I	India-brief history of Indian constitution ,framing-features fundamental rights, duties,directive principles of states, History of Indian National movement,Socio economic growth after independence.
II	Society-Social groups-concepts and types,socialization-concept theory, social control:concept,social problem in contempory India,status and role.
III	The fundamental of Economics-meaning, definition and importance of economics, Logic of choice, central economic problems, positive and normative approaches, economic systems-socialism and capitalism.
IV	Microeconomics-Law of demand and supply, utility approach, indifference curves, elasticity of demand & supply and applications, consumer surplus, Law of returns to factors and returns to scale.
V	Macroeconomics- concept relating to National product National income and its measurement,simple Keynesian theory, simple multiplier, money and banking. Meaning, concept of international trade, determination of exchange rate, Balance of payments.

References:

1. Economics-Lipsey & Chrystal, Oxford Univ.Press,2010
2. Nordhaus, William, Samuelson,Paul-2009-10