COMPUTER NETWORKS

Delay Calculations:

- Propagation delay = $\frac{Distance\ Traveled\ by\ frame\ or\ packet}{Propagaton\ speed}$
- Transmission delay = $\frac{No.of bits\ to\ transfer}{Transmission\ speed\ or\ bandwidth(in\ bps)}$

Channel Utilization:

1. For *Ethernet*, channel utilization, U = $\frac{1}{1+5a}$, where a = $\frac{propagation\ delay}{transmission\ delay}$
   1.1. *for 1- persistent CSMA/CD* LAN 10-1000Mbps speed, simplified channel efficiency is,
   $[Channel\ efficiency] = \frac{T_d}{T_d+\frac{2\tau}{A}}$, where A =$kp(1-p)^{k-1}$, k stations each with probability p to transmit in a contention slot. T_d is the time to transmit a frame. τ is the worst case one way propagation delay.
   1.2. *Efficiency of 802.3 Ethernet at 10Mbps with 512 bit connection*:
   $(Channel\ Efficiency) = \frac{1}{1+\frac{2BLe}{cF}}$, where, B = network bandwidth. L = Cable Length. C = speed of signal propagation. E = optimal number if contention slots per frame. F = frame size
2. For *Token ring* (**release after transmission or early token release**),
   2.1. Channel utilization, U =$\frac{1}{1+\frac{a}{N}}$, where N is the number of stations in the ring.
3. For *Token ring* (**release after reception or delayed token release**),
   3.1. Channel utilization, U =$\frac{1}{1+a}$, where N is the number of stations in the ring.
4. For *unrestricted simplex protocol*: If a frame has d data bits and h overhead bits and channel bandwidth = b bits/sec then,
   4.1. Maximum channel utilization = $\frac{Data\ Size}{Frame\ Size}$ = $\frac{d}{d+h}$
   4.2. Maximum data throughput = $\frac{Data\ Size}{Frame\ Size} \times Bandwidth$ = $\frac{d}{d+h} \times b$
5. For *stop-and-wait*,
   5.1. Channel utilization, U =$\frac{1-p}{1+2a}$, where a = $\frac{propagation\ delay}{transmission\ delay}$ and p is the probability that a frame is in error.
   5.2. Also Maximum channel utilization = $\frac{Time\ to\ transmit\ a\ frame}{Round\ trip\ time\ (R)} \times \frac{d}{d+h}$ = $\frac{d}{b \times R}$

5.3. Maximum data throughput = $\frac{d}{b \times R} \times b = \frac{d}{R}$

6. For Simplex positive acknowledgement with retransmission (PAR) protocol: -
   6.1. Maximum channel utilization and throughput is similar to stop-and-wait protocol when the effect of errors is ignored.
7. For *Sliding Window Protocols* with window size of **w**,
   7.1. Go-Back-N,
      7.1.1. Channel utilization, u

$$= \begin{cases} \frac{1-p}{1+2ap}, & if\ window\ fills\ the\ pipe\ i.e., w \geq 2a+1 \\ \frac{w(1-p)}{(1+2a)(1-p+wp)}, & if\ window\ does\ not\ fills\ the\ pipe\ i.e., w < 2a+1 \end{cases}$$

   7.2. Selective reject,
      7.2.1. Channel utilization ,u $= \begin{cases} (1-p), & if\ window\ fills\ the\ pipe\ i.e., w \geq 2a+1 \\ \frac{w(1-p)}{(1+2a)}, & if\ window\ does\ not\ fills\ the\ pipe\ i.e., w < 2a+1 \end{cases}$
   7.3. Condition for maximum utilization or throughput is:
   $[Time\ to\ transmit\ \boldsymbol{w}\ frames] \geq [Round\ Trip\ Time]$

Throughput Calculations:

Throughput = Channel Utilization × Channel Bandwidth

Signal and Noise Calculations:

1. *Signal to Noise Ratio* (in decibels, **dB**) = $10\log_{10}\frac{S}{N}$,
   a. where S= Signal strength and N = noise strength.
2. *Signal Attenuation* (in decibels, **dB**) = $10\log_{10}\frac{Transmitted\ Power}{Received\ Power}$,

Data Rate and Channel Capacity Calculations:

1. *Nyquist Theorem*: Maximum data rate = $2H\log_2 V$ bits/sec, where H is bandwidth in hertz (Hz) and V is number of levels.
2. *Shannon's theorem*: Channel capacity = $H\log_2\left(1+\frac{S}{N}\right)$ bits/sec, where H is bandwidth in hertz (Hz). (Note: here $\frac{S}{N}$ is **not** in decibels).

Baud rate: A baud is the number of changes per second in the signal.

- For Manchester encoding, **baud rate = 2 × bit-rate**

MAC Sub layer:

Static channel allocation in LANs and MANs.

If C = channel capacity in bps

λ = arrival rate of frames (frames/sec)

$\frac{1}{\mu}$ = no. of bits per frame, then

Mean time to delay, T = $\frac{1}{\mu C-\lambda}$,

Now, if the channel is divided into N sub channels each with capacity $\frac{C}{N}$ and arrival rate or input rate on each of the N channels is $\frac{\lambda}{N}$ then,

T'(fdm) = $\frac{1}{\mu\frac{C}{N}-\frac{\lambda}{N}} = \frac{N}{\mu C-\lambda}$

Dynamic Channel Allocation:

$\begin{bmatrix} Probability\ of\ a\ frame \\ being\ generated\ in\ a\ period\ of\ length\ \Delta t \end{bmatrix} = \lambda\Delta t$, where λ is the arrival rate of frames.

Multiple access protocol:

Pure ALLOHA protocol

- Infinite senders assumed.
- Poisson distribution with mean rate **S** frames per frame time
- Combined frame rate with retransmission **G** frames per frame time.
- '**t**' is the time required to transmit a frame.
- In multiple access protocol, a frame is successful if no other frames are transmitted in the vulnerable period.
- Probability of **k** frames being generated during a frame transmission time:
  $P_k = \frac{G^k e^{-G}}{k!}$
- Hence, probability of zero frames in 2 frame periods is, $P_0 = e^{-2G}$
- Therefore, for pure ALLOHA,
  - Mean rate $S = GP_0 = Ge^{-2G}$ which becomes maximum at G = ½,
  Max(S) = $\frac{1}{2e}$ = 0.184 = 18.4% throughput.
- Vulnerability period in pure ALLOHA: For successful frame transmission, no other frame should be on the channel for vulnerability period equal to twice the time to transmit one frame. That is,
  $\begin{pmatrix} Vulnerability\ period \\ in\ case\ of\ PURE\ ALLOHA \end{pmatrix} = 2t$, where t is the time to transmit one frame.

Slotted ALLOHA protocol

- Time is divided into discrete frame slots.
- A station is required to wait for the beginning of the next slot to transmit a frame.
- Vulnerability period is halved as opposed to pure ALLOHA protocol. That is,
  $\begin{pmatrix} Vulnerability\ period \\ in\ case\ of\ SLOTTED\ ALLOHA \end{pmatrix} = t$, where t is the time to transmit one frame.
- Probability of **k** frames being generated during a frame transmission time:
  $P_k = e^{-G}(1-e^{-G})^{k-1}$

- Hence, probability of zero frames in 1 frame period is, $P_0 = e^{-G}$
  - Mean rate $S = GP_0 = Ge^{-G}$ which becomes maximum at G = 1,
  Max(S) = $\frac{1}{e}$ = 0.368 = 36.8% throughput.
- Expected number of retransmission, $E = e^G$.

PPP

- In **Point to Point Protocol** (PPP), number of channels grows as square of the number of computers. That is, $\begin{pmatrix} Number\ of\ Channels\ or\ links \\ for\ n\ computers \end{pmatrix} = \frac{n(n-1)}{2}$

Binary Exponential Backoff Algorithm:

- After **i** collisions wait a random number of slots between 0 and $2^i - 1$ with a maximum of 1023. (**Note:** *After 16 collisions, failure is reported to the higher layers*.)

Minimum frame size for IEEE 802.3 (Ethernet) frame = 64 bytes.

ROUTING ALGORITHMS	STATE
Shortest Past algorithm	Non-Adaptive (or Static)
Flooding Algorithm	Non-Adaptive (or Static)
Flow Based Routing Algorithm, It uses the formula, Delay time, T = $\frac{1}{\mu C - \lambda}$, where C= channel capacity, $1/\mu$ = mean packet size in bits, and λ = mean number of arrivals in packets/sec	Non-Adaptive (or Static)
Distance Vector Routing (DVR) • Based on Bellman-Ford Algorithm and the Ford-Fulkerson Algorithm. • It suffers from the "Count to infinity problem" • Exchange information of the entire network with its neighbors.	Adaptive Algorithm (or non-static)
Link State Routing Algorithm (LSR) • Discovers its neighbors and construct a packet telling all it has just learned and send this packet to all other routers.	Adaptive algorithm (or non-static)
Hierarchical Routing Algorithm • For N router subnet, the total number of levels = $\ln N$ • And each router will have $e \times \ln N$ number of entries in their routing tables.	Adaptive Algorithm (or non-static)
Broadcast Routing Algorithm	Adaptive Algorithm (or non-static)

- Congestions deals with wires and routers while flow deals with hosts.
- **Traffic Shaping**:
  - Leaky Bucket Algorithm (If the bucket overflows, then packets are discarded).
  - Token Bucket Algorithm (causes a token to be generated periodically).
- Congestion control through Load Shedding may lead to deadlock and unfairness.

- The size of data portion of the datagram is = [Total length] – [size of header]
- Maximum length of the datagram = 65535 bytes.

Datagram format:

Total length	16 bits
Header length	4 bits
Flag length	3 bits
Type of service	8 bits
Identification bits	16 bits
Fragment offset	13 bits
Time to Live	8 bits
Protocol version	8 bits
Header Checksum	16 bits
Source Address	32 bits
Destination Address	32 bits

- In Layer 2 of OSI model (Data link layer), destination field appears before source field where as in layer 3 (Network layer), the ordering is opposite.

IP class addressing:

Class Name	Starts with	Range
Class A	0	0-127
Class B	10	128-191
Class C	110	192-223
Class D	1110	224-239
Class E	11110	240-255

- Internet addresses refer to network connections rather than hosts. (For example, Gateways have two or more network connections and each interface has its own IP address). Thus, Physical (or Ethernet) address is constant (fixed) but Network (or IP address) may change.

Transport Layer:

To cope with the widely varying delays, TCP maintains a dynamic estimate of the current RTT (Round Trip Time) calculated this way:

- When sending a segment, the sender starts a timer.
- Upon receipt of an acknowledgement, it stops the timer and record the actual elapsed delay M between sending the segment and receiving its acknowledgement.
- Whenever a new value M for the current RTT is measured, it is averaged into a smoothed RTT depending on the last measured value as follows:
  - **New RTT = a (Old RTT) + (1-a)(Current RTT M),** where **a** is known as the smoothing factor and it determines how much weight the new measurement carries. When **a** is 0, we simply use the new value and when **a** is 1 we ignore the new value. Typically, the value of **a** lies between 0.8 and 0.9
- **TCP** can provide reliable service where UDP cannot as they choose to use. IP in two different modes of service provided by IP either reliable or connectionless.

- A transparent router connects a LAN to a WAN.
- TCP does not have in-built mechanism to distinguish between a timeout owing to error and that owing to congestion.
- During early start phase, the congestion window inside TCP grows Exponentially, whereas after that it grows linearly.
- If two client try to connect to the same TCP service, then they can
- Private network address enables reuse of IP addresses, thereby aiming to reduce the IP shortage problem.

CLIENT SERVER MODEL:

SERVER	CLIENT
Socket	Socket
Bind	Connect
Listen()	Send()
Accept()	Receive()
Receive()	Close()
Send()	

- Terms connected with RPCs(Remote Procedural Calls):
  - Marshalling, Shunt
  - Client, Skelton
  - Procedure call, server
- Terms not-connected with RPCs:
  - Socket, connect.
- While trying to access a webpage www.facebook.com, a user gets the "http" error message "could not resolve host name". What could be the cause of this problem?
  - The local DNS is not running.
- In TCP "SYN" flag is used for connection establishment.
- **A** wants to send a group message to a number of people. How can it be ensured that the message actually came from **A** and not from someone else?
  - **A** encrypts via its **own private key**; and the group decrypts via **A's public key**.
- The number of bits used for IPV6 addressing is 128 bits whereas for Ethernet address 48 bits are used.
- If **n** is the number of bits used to represent the frame sequence number then:

Name of sliding window protocol	Sender window size	Receiver window size
Go-back-N	$2^n - 1$	1

Selective Repeat	2^{n-1}	2^{n-1}

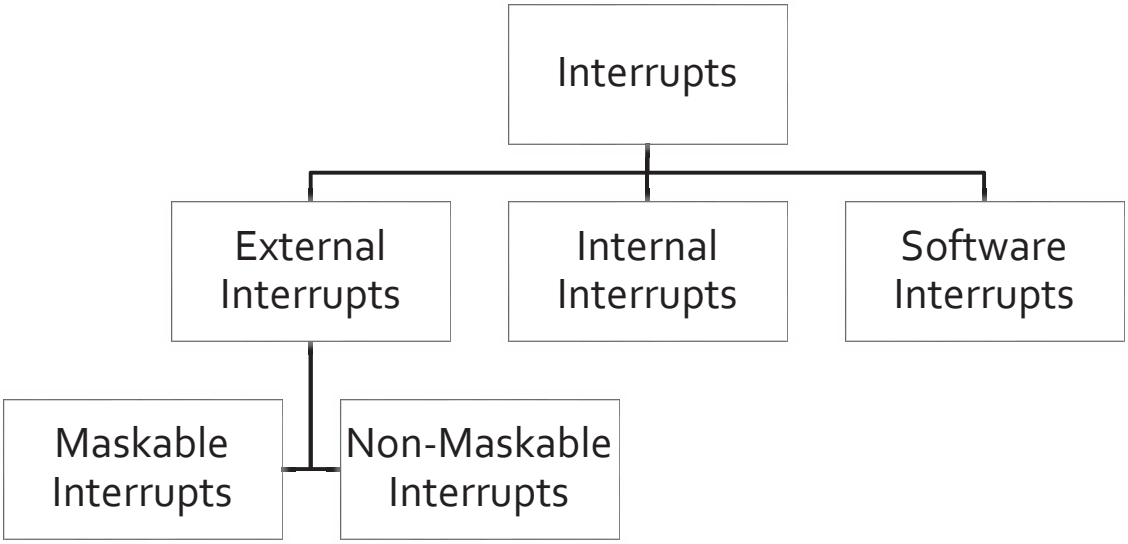
- The maximum burst rate at which network can handle the traffic is given by,
  $\frac{C}{M-R}$, where C = Capacity of bucket(in bits); M = Network rate(e.g., token ring with 10 Mbps network LAN); R = arrival rate(e.g., rate of entering into the bucket).
- In any sliding window protocol,
  - $The\ optimal\ window\ size = RTT \times Channel\ Bandwidth$,
- $Residual\ Error\ Rate = \frac{Number\ of\ lost\ or\ garbled\ messages}{Total\ Sent}$,
- $Ping$ Works on IP Layer / Network layer.

Devices and their OSI Layer:

GATEWAY	Application Layer, Presentation Layer, Session Layer, Transport Layer
Router	Network Layer
Bridges and Switches	Data Link Layer
Repeaters, Hubs, Amplifiers, Multiplexers	Physical Layer

- A group of $(2^n - 1)$ routers are interconnected in a centralized binary tree, with a router at each node. Router **I** communicate with router **J** by sending a message to the root of the tree. The root then sends the message back down to J, then
  - $\left(\begin{array}{c} The\ mean\ number\ of\ hops \\ per\ message\ for\ large\ n \\ \boldsymbol{Assuming\ that\ all\ router\ pairs} \\ \boldsymbol{are\ equally\ likely} \end{array}\right) = \frac{2(2^n \times n - 2(2^n - 1))}{2^n - 1}$,

COMPUTER ORGANIZATION AND ARCHITECTURE



- Maskable interrupts are enabled and disabled by executing instructions such as EI or DI. If the computer's interrupts is disabled, the computer ignores the maskable interrupt.
  - Interrupt is disabled -> Interrupt flag bit = 1
  - Interrupt is enabled -> Interrupt flag bit = 0
- The Non-Maskable interrupt has higher priority than the maskable interrupt.
- If both maskable and non-maskable interrupts are activated at the same time, the processor will service the non-maskable interrupt first.
- Non- maskable interrupts are typically used in power failure interrupts.

PIPELINE

- $\begin{pmatrix} Clock\ Period\ of \\ pipeline \end{pmatrix} = MaxDelay(T1, T2, T3, \dots) + L$, where, T_i= pipeline segment delay; L = Latch delay.
- $(Speed\ Up\ of\ Pipeline) = \frac{Time\ Taken\ to\ perform\ the\ Task\ without\ Pipeline}{Time\ Taken\ to\ perform\ the\ Task\ with\ Pipeline}$
- $Throughput\ of\ pipeline = \frac{Total\ number\ of\ tasks\ to\ perform}{Total\ time\ to\ perform\ the\ task\ with\ Pipeline}$
- $Efficiency\ of\ Pipeline = \frac{Speed\ up\ of\ pipeline}{Maximum\ speed\ up\ of\ pipeline}$

- $Maximum\ speed\ up\ of\ pipeline = Total\ number\ of\ pipe\ segments\ in\ the\ pipeline$
- **Hazards of Pipeline**:
  - Branch Instruction: For each branch instruction additional (n-1) pipeline clocks are required; where n is the number of segments in the pipeline.
  - Data dependency
  - Resource Conflicts.
- $\begin{pmatrix} Total\ number\ of\ clocks\ required\ to\ proess \\ a\ stream\ of\ S\ instructions \end{pmatrix} = (n + S - 1) + SC(n - 1)$, where S= no. of instructions; n = no. of pipe segments; C= Probability for an instruction to be a conditional branch.
- Cache works on the principle of locality.
- Each process has its own virtual address space.
- The unit block of data for transferring between disk and memory is called a page. (Typically 4kB)
- TLB is actually a cache for the page table.
- Virtual memory uses write back method because frequent access to disk is very slow.
- A DRAM consists of 1 transistor and 1 capacitor.
- An interrupt in which device supplies the interrupt requests as well as its address is called Vectored Interrupt.
- When INTER is encountered, the processor branches to the memory location which is-
  - Determined by the RST N instruction given by the IO device.
  - Determined by the Call address instruction given by the IO device.
- In 8085 microprocessor,
  - XCHG: Exchange the content of the HL and DE register pair respectively.
  - DAD H: Add HL pair with HL pair and store the result in HL pair.
- Thrashing implies excessive page IO
- In 8085, a stack pointer is a 16-bit register that points to stack.
- $(Read\ Stall\ Cycles) = Read\ \times Read\ miss\ rate\ \times Read\ miss\ rate \times Read\ miss\ penalty$,
- $(Write\ Stall\ Cycles) = writes \times write\ miss\ rate \times write\ miss\ penalty + write\ buffer\ stall$
- $(Memory\ Stall\ Cycles) = Memory\ Access \times Cache\ Miss\ rate \times Cache\ Miss\ Penalty$
- $(CPU\ Time) = Ic\ \times \left(CPI + \frac{Memory\ clock\ cycle}{Intruction}\right) \times Clock\ cycle\ time$, where Ic= number of instructions; CPI=CPU clock cycles;
- PARAM is a parallel computer.
- ALE is a pin of an 8085 microprocessor that is used to the latch the 8 bits of address lines AD7-AD0 into an external latch.
- The magnetic bubble memory has an access time of 30μs.
- The magnetic material that is normally used in bubble memories is GARNET.

- For RELATIVE ADDRESSING:
  - Effective address = Program counter + Address part of instruction.
    - For example: 2 word instruction, I = JMP 65 stored in 2000 in decimal.
      - So EA = PC + Address part; EA = 2002 + 65 =2067 in decimal.
- $(CPU\ time\ on\ machine\ 1) = \frac{CPU\ clock\ cycles\ of\ the\ program}{Clock\ rate\ for\ M1}$,
- $(Speed\ Up) = \frac{Performance\ with\ enhancement}{Performance\ without\ enhancement} = \frac{Execution\ time(old)}{Execution\ time\ (new)}$
- Memory Chip notation:$(Memory\ Depth) \times (Memory\ Width)$;e.g., 16M * 8 = 128 Mb
- $(Memory\ Density) = (Memory\ Depth) \times (Memory\ Width)$
- Chip notation:$(Memory\ Depth\ per\ bank) \times (Memory\ Width) \times (number\ of\ banks)$
- For memory access:
  - Cycle Time = Latency Time + Transfer Time.
- $(Utilization\ Factor) = \frac{No.of\ clock\ cycles\ in\ pipeline}{No.of\ clock\ cycles\ in\ non-pipeline}$
- $Average\ Instruction\ Execution\ Time = (Average\ CPI) \times (Clock\ cycles\ time)$
- Maximum Forbidden latency: $\leq (n-1)$; where n = number of columns in the reservation table.
- A SIMD (Single instruction multiple data) is characterized by, $C =< N, F, I, M >$ notation.
- $\binom{\boldsymbol{Average\ Memory}}{\boldsymbol{Access\ Time}} = (\boldsymbol{Hit\ time}) + (\boldsymbol{Miss\ Rate}) \times (\boldsymbol{Miss\ Panelty})$
- $\binom{\boldsymbol{Effective\ Memory}}{\boldsymbol{Access\ Time}} = (\boldsymbol{Hit\ Rate})(\boldsymbol{Hit\ time}) + (\boldsymbol{Miss\ Rate}) \times (\boldsymbol{Miss\ Panelty})$
- $\binom{\boldsymbol{Average\ Instruction}}{\boldsymbol{Execution\ time\ in\ a\ k-stage\ pipeline}} = \binom{\boldsymbol{Clock\ cycle\ time}}{\boldsymbol{of\ pipeline}} + (\boldsymbol{Stall\ Time}) \times (\boldsymbol{Stall\ Frequency})$
- How many bits will be required to store a decimal number containing (a) 3 digit (b) 4 digit (c) 96 digits
  - Key formula is $\boldsymbol{No.of\ possibilities} = (\boldsymbol{base})^{\boldsymbol{No.\ of\ digits}}$
  - (No. of possibilities in base 2) = (No. of possibilities in base 10) = (No. of possibilities in any base)
  - Let N be the required number of bits and n be the number of decimal digits given, then
  - $2^N = 10^n \rightarrow N = \lfloor \log_2 10^n \rfloor + 1$; so for 96 decimal digit number we would need $N = \lfloor \log_2 10^n \rfloor + 1 = \lfloor \log_2 10^{96} \rfloor + 1 = 318 + 1 = 319\ bits$. in binary number system
- **For page table:**
  - $\binom{No.\ of\ comparators}{required} = (No.\ of\ bits\ in\ the\ line\ offset\ or\ index\ field\ of\ CPU\ address)$
  - $\binom{Bit\ size\ of}{each\ comparator} = (No.\ of\ bits\ the\ TAG\ field\ of\ the\ CPU\ address)$

- $\begin{pmatrix} Maximum\ number\ of\ files \\ in\ a\ file\ system \end{pmatrix} = \frac{\boldsymbol{Directory\ Size}}{\boldsymbol{Dictionary\ Entry\ Size}}$

- ***Amdahl's Law***:

  $\frac{\mathbf{1}}{(\mathbf{1}-\boldsymbol{P})+\frac{\boldsymbol{P}}{\boldsymbol{N}}}$, P is the proportion of a program that can be made parallel (i.e., benefit from parallelization); (1 − P) is the proportion that cannot be parallelized (remains serial).

THEORY OF COMPUTATION

- If an NFA contains N no. of States then the corresponding DFA will contain maximum of 2^N states.
- **Identities for regular expressions:**
  - $\emptyset + \boldsymbol{R} = \boldsymbol{R} + \emptyset = \boldsymbol{R}$
  - $\emptyset \boldsymbol{R} = \boldsymbol{R}\emptyset = \emptyset$
  - $\boldsymbol{\varepsilon R} = \boldsymbol{R\varepsilon} = \boldsymbol{R}$
  - $\boldsymbol{\varepsilon}^* = \boldsymbol{\varepsilon}$ & $\emptyset^* = \boldsymbol{\varepsilon}$
  - $\boldsymbol{R} + \boldsymbol{R} = \boldsymbol{R}$
  - $\boldsymbol{R}\boldsymbol{R}^* = \boldsymbol{R}^*\boldsymbol{R} = \boldsymbol{R}^+$
  - $((\boldsymbol{R}^*)^*) = \boldsymbol{R}^*$
  - $\boldsymbol{R}^*\boldsymbol{R}^* = \boldsymbol{R}^*$
  - $(\boldsymbol{P} + \boldsymbol{Q})^* = (\boldsymbol{P}^*\boldsymbol{Q}^*)^*$; Where P and Q are regular expression.
  - $\boldsymbol{R}(\boldsymbol{P} + \boldsymbol{Q}) = \boldsymbol{R}\boldsymbol{P} + \boldsymbol{R}\boldsymbol{Q}$ & $(\boldsymbol{P} + \boldsymbol{Q})\boldsymbol{R} = \boldsymbol{P}\boldsymbol{R} + \boldsymbol{Q}\boldsymbol{R}$
  - $\boldsymbol{P}(\boldsymbol{Q}\boldsymbol{P})^* = (\boldsymbol{P}\boldsymbol{Q})^*\boldsymbol{P}$
- **Arden's theorem:**
  - If P & Q are two regular expressions over an alphabet S such that P does not contain ε then the following equation:
    - $\boldsymbol{R} = \boldsymbol{Q} + \boldsymbol{R}\boldsymbol{P}$ in R has a unique solution(only one solution) i.e., $R = QP^*$
- Some important results:
  - If L1 is DCFL and L2 is regular, then L1∪L2 is also DCFL
  - If L1 is DCFL and L2 is regular, then L1∩L2 is also DCFL
  - Every regular language is DCFL
  - The union of DCFL & regular language is DCFL
  - $DPDA \not\equiv NPDA$
  - $DPDA \subset NPDA$
  - $Power\ of\ DPDA < Power\ of\ NPDA$
  - $\begin{pmatrix} Prmitive\ Recursive \\ function \end{pmatrix} \subset \begin{pmatrix} Total\ Recursive \\ function \end{pmatrix} \subset \begin{pmatrix} Partial\ Recursive \\ function \end{pmatrix}$
  - All RE languages are TM enumerable.
  - The power set of an infinite set is not countable.

- Not all languages are recursively enumerable.
- Recursive languages are closed under complementation but recursively enumerable languages are NOT closed under complementation.
- The complement of a CFL may or may not be closed under complementation.

- The minimum number of states in DFA accepting strings containing number of 0's divisible by P and 1's divisible by Q = $P \times Q$
- The number of possible DFA's with N states and M input symbol = $2^N N^{N \times M}$
- For the strings over some alphabet $\sum$, the following are primitive regular expressions:
  - X, for each $x \in \sum$
  - $\lambda, the\ empty\ string$
  - $\phi, indicating\ number\ of\ string.$
    - Thus if $|\sum| = n,$ then there are $(n+2)$ primitive regular expressions defined over $\sum$.
    - Every primitive regular expression is a regular expression.
- For a set A of n distinct elements:
  - Number of symmetric and reflexive relations = $2^{\frac{n(n-1)}{2}}$
  - Number of symmetric relations = $2^{\frac{n(n+1)}{2}}$
  - Number of reflexive relations = $2^{n(n-1)}$
  - Number of irreflexive relations = $2^{n(n-1)}$
  - Number of transitive relations = 1,2,13,171,3994 … for 1 ,2 ,3 ,4 , 5 ,.. elements
  - Number of anti-symmetric relations = $2^n 3^{\frac{n(n-1)}{2}}$
  - Number of reflexive and anti-symmetric relations = $3^{\frac{n(n-1)}{2}}$
  - Number of equivalence relations = 1,2,5,15,52,203… for 1,2,3,4,5,6,… elements. (i.e., the BELL number).
  - Number of all possible relations = 2^{n^2}

DATA STRUCTURE AND ALGORITHMS

C operators in order of precedence (highest to lowest). Their associativity indicates in what order operators of equal precedence in an expression are applied.

Operator	Description	Associativity
()	Parentheses(function call)	Left-to-Right
[]	Brackets (array subscript)	
.	Member selection via objet name	
->	Member selection via pointer	
++ --	Postfix increment / decrement	
++ --	Prefix increment / decrement	Right-to-Left
+ -	Unary Plus / minus	
! ~	Logial negation/bitwise complement	
(type)	Cast (convert value to temporary value of type)	
*	Dereference	
&	Address (of operand)	
sizeof	Determine size in bytes on this implementation	
* / %	Multiplication / division / modulus	Left-to-Right
+ -	Addition / Subtraction	Left-to-Right
<< >>	Bitwise shift left, Bitwise shift right	Left-to-right
< <=	Relational less than/ less than or equal to	Left to right
> >=	Relational greater than/ greater than or equal to	
== !==	Relational is equal to/is not equal to	Left to right
&	Bitwise AND	Left to right
^	Bitwise exclusive OR	Left to right
|	Bitwise inclusive OR	Left to right
&&	Logical AND	Left to right
||	Logical OR	Left to right
?:	Ternary conditional	Right to left
=	Assignment	Right of left
+= -=	Addition/subtraction assignment	
*= /=	Multiplication / division assignment	
%= &=	Modulus / bitwise AND assignment	
^= |=	Bitwise exclusive / inclusive OR assignment	
<<= >>=	Bitwise shift left / right assignment	
,	Comma (separate expression)	Left to right

- Radix sort is a **non-comparative integer sorting algorithm**.

- Worst case time complexity = O(kN); N number of numbers each with k digits.
- Worst case space complexity = O(k + N)

- Insertion sort is stable and inplace sorting.
  - Best case time complexity = O(n) and
  - worst case time complexity = O(n^2)
- Binary Insertion sort
  - Best case time complexity = O($\mathrm{nlog}\, n$)
  - Worst case time complexity = O(n^2)
- For a look up in a B+ tree, we traverse from root node to leaf node. If there are K search keys and number of pointers per node (or degree) is P, then for a look up,

$$\begin{pmatrix} The\ path\ is \\ no\ longer\ than \end{pmatrix} = \frac{\left\lceil \log\left\lfloor \frac{K+1}{2} \right\rfloor \right\rceil}{\left\lceil \log\left\lceil \frac{P}{2} \right\rceil \right\rceil},$$

- Number of articulation points in a tree = number of internal nodes.
- For a full K-ary tree,
  - $Total\ number\ of\ nodes = K \times (number\ \ of\ internal\ nodes) + 1$
- Formula that computes the address LOC(A[J,K]) of A[J,K]
  - Column-major order:
    - $LOC(A[J,K]) = BASE(A) + w[M(K-1) + (J-1)]$
  - Row – major order:
    - $LOC(A[J,K]) = BASE(A) + w[(K-1) + N(J-1)]$
- where Array size = (M,N); M= number of rows; N= number of columns;

Relationships and definitions:

Asymptotic form	Relationship	Definition
$fn \in \theta(gn)$	$fn \equiv gn$	$0 < \lim_{n\to\infty} \frac{fn}{gn} < \infty$
$fn \in O(gn)$	$fn \leq gn$	$0 \leq \lim_{n\to\infty} \frac{fn}{gn} < \infty$
$fn \in \Omega(gn)$	$fn \geq gn$	$0 < \lim_{n\to\infty} \frac{fn}{gn}$
$fn \in o(gn)$	$fn < gn$	$\lim_{n\to\infty} \frac{fn}{gn} = 0$
$fn \in \omega(gn)$	$fn > gn$	$\lim_{n\to\infty} \frac{fn}{gn} = \infty$

- o(gn) $\wedge\, \omega(gn)$ is the empty set. $((a < b) \cap (a > b))$

- For any real constant a and b, where b>0
  - $(n+a)^b = \theta(n^b)$
- **Complexity ordering:**
  - $O(1) < O(\log_2 n) < O(n) < O(n^2) < \cdots < O(n^k) < O(2^n)$
- So space requirement for an inplace sort is O(1).
- Bubble sort:
  - We need to make a maximum of (n-1) passes for n-input elements.
  - $\begin{pmatrix} Total\ number\ of\ comparisons \\ in\ bubble\ sort\ after\ k-iteration\ of \\ outer\ loop \end{pmatrix} = \frac{2kn-k^2-k}{2}$,
  - The average number of iterations of outer loop = O(n)
  - Total number of comparisons required = $\frac{n^2}{2}$
  - Total number of sqaping required for arranging n elements in the sorted order by using bubble sort = $\frac{3n^2}{4}$
  - And hence the average time complexity = O(n^2)
  - Space complexity of bubble sort = O(1)
  - The number of interchanges an never be greater than the number of comparisons.
  - If the file is already sorted then the time taken by bubble sort is order of O(n), as it will require only one pass of (n-1) comparisons.
- Quick sort:
  - Best and average case time complexity = O(n logn)
  - Worst case time complexity = O(n^2)
  - Worst case space complexity = O(n)
- Insertion Sort:
  - Best case:
    - Time complexity = O(n)
  - Average and Worst case:
    - Time complexity = O(n^2)
  - Total number of comparisons are = $\sum_{j=2}^{n-1} \frac{j+1}{2} = \frac{n^2+n-2}{4}$
- Selection Sort:
  - Best, average, and worst case time complexity = O(n^2)
  - Worst case space complexity = O(n) {total} and O(1) {auxiliary}

  - Selection sort has the minimum number of swaps.
- SHELL sort:
  - It is also known as diminishing increment sort.
  - Time complexity = O($n(\log n)^2$)
  - If n is an exact power of 2, then average running time is O($n^{\frac{3}{2}}$)
- Merge Sort:
  - Time Complexity = O(n logn)
  - Space complexity of Merge sort = O($\log n$)
- Binary Search:
  - Maximum number of comparisons is approximately $\log_2 n$
- Interpolation search:
  - Time complexity = log (log n)
  - Worst case behaves like linear search = O(n)
- Robust interpolation search worst complexity = $(\log n)^2$
- **Hashing:**
  - If n: no. of elements; m: no. of slots, and $\boldsymbol{\alpha = \frac{n}{m}}$
    - Resolve by chaining unsuccessful search takes $\theta(1+\alpha)$
    - Successful search takes $\theta\left(2+\frac{\alpha}{2}+\frac{\alpha}{2n}\right)$
- Double hashing:
  - Here we use two different hash functions as
    - h(k,i) = (h1(k) + ih2(k)) mod m
    - Initial probe is to position T(h1(k))
  - Here $\theta(m)^2$ probe sequences are used.
- In an open addressing scheme, the expected number of probes in an unsuccessful search is at most $\frac{1}{1-\alpha}$ assuming uniform hashing.
  - And expected number of probes in a successful search is at most $\frac{1}{\alpha}\ln\frac{1}{1-\alpha}$
- The total number of nodes in a complete binary tree = $2^{d+1}-1$; where d is the depth.
  - Number of leaf nodes = 2^d
  - Number of non-leaf nodes = 2^d-1
- Time complexity of matrix chain multiplication = O(n^3)
- Balance tree:
  - AVL tree

  - B-Tree.
- Kruskal's Algorithm:
  - The total time taken by this algorithm to find the minimum spanning tree is $O(E \log_2 E)$
  - But the time complexity, if edges are not sorted is $O(E \log_2 V)$
  - Where E: number of edges; V: number of vertices.
- Prim's Algorithm:
  - The time complexity for Prims algorithm is $O(E \log_2 V)$
- The depth of a complete binary tree with n nodes:
  - $Depth = \log_2 n + 1$
- The worst case height of AVL tree with n nodes = $1.44 \log n$
- Number of trees (not only binary trees) possible with n- nodes = $2^n - n$
- For a complete n-ary tree having either 0 or 1 sons. If k is the number of internal nodes of a complete n-ary tree, then the number of leaves in it is given by = $k(n-1)+1$
- Heapify() call takes $O(\log n)$ times.
- Heap Sort:
  - The running time is $O(n \log n)$
  - A tighter bound is $\theta(n)$
- Number of different unlabelled binary trees possible with n-nodes = $\frac{1}{n+1}\binom{2n}{n}$
- Number of different labeled binary tree possible with n-nodes = $\frac{1}{n+1}\binom{2n}{n}(n!)$
- The height of a red black tree with n- internal nodes is = $(2 \log_2 n - 1)$
- Suppose we use a hash function h to hash n distinct keys into a array T of length m. Assuming simple uniform hashing:
  - The expected number of collisions = $\frac{n(n+1)}{2m}$
- Number of moves required to solve the Tower of Hannoi Problem = $2^n - 1$; where n is the number of disks.
- Dijkstra's algorithm:
  - With V vertices and E edges, Time complexity = $O(E + V \log V)$
- Bellman Ford Algorithm:
  - With V vertices and E edges; Time complexity = O(EV)
- Let T(l) and T(r) denote the left and right subtrees of a binary tree T. If T has n nodes, then the
  - Total Path Length, P(T) = P(l)+P(r) + n-1 ; where P(l) is the path length of left subtree and P(r) is the path length of right subtree.

- Different implementation of Hashing and their time complexity:

Implementation	Insert	Search
Direct addressing	O(1)	O(1)
Ordered addressing	O(N)	O(log N)
Ordered List	O(N)	O(N)
Unordered array	O(1)	O(N)
Unordered List	O(1)	O(N)

- Chromatic number of a bipartite graph = 2
- Chromatic number of a planar graph ≤ 4
- We can find the coloring of n vertices in O(2^n)
- Time complexity of DFS with V vertices and E edges = O(V + E)
- Time complexity of BFS with V vertices and E edges = O(V + E)
- Total number of BST(binary search trees) with n distinct keys = $\frac{1}{n+1}\binom{2n}{n}$
- In complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is = n(k – 1) + 1
- For a hash table of size N,
  - Probability of inserting K integers with no collision = $e^{\frac{-K(K-1)}{2N}}$
  - Thus probability of collision for inserting K integers = $1 - e^{\frac{-K(K-1)}{2N}} \approx \frac{K(K-1)}{2N} \approx \frac{K^2}{2N}$
- Total number of stack permutations of a n distinct element is = $\frac{1}{n+1}\binom{2n}{n}$
- Total number of invalid permutations = $n! - \frac{1}{n+1}\binom{2n}{n}$
- Time complexity of Bankers algorithm = O (mn^2); where m= number of resources and n= number of processes.
- An n-dimensional hypercube (Q_n) is simple undirected graph with 2^n vertices. The vertices of (Q_n) are bit-strings of length n. Two vertices of (Q_n) are connected by an edge only if u & v differs in exactly one bit position.
  - (Q_n) is Regular bipartite graph.
  - Degree of each vertex =2 => is bipartite.
  - Number of edges = $n2^{n-1}$
- Floyd Warshall algorithm has time complexity = O(V^3)
- Fibonacci series recurrence relation regardless of any initial condition is

- T(n) = T(n-1) + T(n-2) + n = $2^n + n2^n$
  - Best case = $\Omega(2^n)$
  - Worst case = $O(n2^n)$

- Fibonacci series recurrence relation regardless of any initial condition is
  - T(n) = T(n-1) + T(n-2) + K (constant)
    - Time complexity in all cases = $O(2^n)$

- **MASTER METHOD**
  - It is applicable to only limited classes of recurrences which are in the form of,
    - $\boldsymbol{T(n) = aT\left(\frac{n}{b}\right) + fn}$; and a >=1 and b >1 and fn is asymptotically positive.{that is, fn>0 for n>=n0}
    - Compare fn with $n^{\log_b a}$
      - Case 1: If fn = $O\left(n^{\log_b a - \varepsilon}\right)$, for some $\varepsilon > 0$
        - **Tn = $\boldsymbol{\theta(n^{\log_b a})}$**
      - Case 2: If fn = $\theta\left(n^{\log_b a} log^k n\right)$, for some $k \geq 0$
        - **Tn = $\boldsymbol{\theta(n^{\log_b a} log^{k+1} n)}$**
      - Case 3: If fn = $\Omega\left(n^{\log_b a + \varepsilon}\right)$, for some $\varepsilon > 0$ and $af\frac{n}{b} \leq (1 - \varepsilon')fn$
        - **Tn = $\boldsymbol{\theta(fn)}$**

- For example, Find the time complexity of the following recurrence relation:
  - $\boldsymbol{Tn = 2T\left(\frac{n}{2}\right) + n\log n}$
    - $\boldsymbol{n^{\log_2 2} = n}$; so fn =nlogn = $\theta(nlogn)$
    - Hence by case 2 of master theorem,
      - Tn = $\boldsymbol{\theta(n^{\log_b a} log^{k+1} n)}$ = $\boldsymbol{\theta\left(nlog^{1+1} n\right) = \theta(nlog^2 n)}$

- Finding complexity by Substitution method for the following recurrence relation:
  - $\boldsymbol{Tn = T(\sqrt{n}) + C1a\ const.}$
    - Let n = 2^m
    - So T(2^m) = T($2^{\frac{m}{2}}$) + C1
    - Let T(2^m) = S(m)
      - S(m) = S($\frac{m}{2}$) + C1 = O(log m) = O(log log n)

DATABASE MANAGEMENT SYSTEM

- Cross Product:
  - If n1 tuples in R and if n2 tuples in S
  - Then, in $P = R \times S$, $(n1 \times n2)$ tuples will be present
- Performance measures on disks:
  - Access time: The time it takes from when a read or write request is issued to when data transfer begins. It consists of:
    - **SEEK TIME:** The time it takes to reposition the arm over the correct track.
      - $\binom{\boldsymbol{Average\ seek}}{\boldsymbol{time}} = \frac{1}{2}\ \boldsymbol{of\ the\ worst\ case\ seek\ time}$
    - **ROTATIONAL LATENCY:** The time it takes to reposition the arm over the correct track.
      - $(\boldsymbol{Average\ Latency}) = \frac{1}{2}\ \boldsymbol{of\ the\ Worst\ Case\ Latency}$
- Lock modes:
  - Read lock – shared lock (s); write lock – Exclusive lock (X)

	R	W
R	**OK**	**WAIT**
W	**WAIT**	**WAIT**

OPERATING SYSTEM

- Next CPU burst predicted is given by:
  - $Tn = \alpha \times tn + (1 - \alpha)T(n - 1)$; $0 \leq \alpha \leq 1$ and tn is current CPU burst, T(n-1) is the past predicted value, and Tn is the new predicted value.
- For n fork() calls $2^n - 1$ child processes will be created.
- Round Robin Algorithm:
  - If there are n processes in ready queue and time quantum is q, then each process gets $\frac{1}{n}$ of the CPU time in chunks of at most q time units.
  - Each process must wait no longer than (n-1)q time units until next time quantum.
  - Turnaround time also depends on the size of the time quantum.
- If waiting time or fraction of each process is P and n is the number of processes, then
  - CPU utilization = $1 - P^n$
  - And probability that N processes will all wait at the same time = P^N
- Unix Inode:
  - Suppose there are 12 direct blocks and 1 indirect block and 1 double indirect block and 1 triple indirect block then the maximum size of process supported by inode is
    $= \left[12 + \frac{BS}{ES} + \left(\frac{BS}{ES}\right)^2 + \left(\frac{BS}{ES}\right)^3\right] \times BS$; where BS = block size; ES = entry size (or block pointer size).

DIGITAL ELECTRONICS

- R's complement:
  - N: a +'ve number in base r with an integer part of n digits, then
    - R's complement of N = $r^n - N$ for N ≠ 0 and 0 for N = 0;
- R-1 complement:
  - N: a +'ve number in base r with an integer part of n digits and a fraction part of m digits, then
    - (R-1)'s complement of N = $r^n - r^{-m} - N$;
- Operator Precedence:
  - () > NOT > AND > OR
- N bit Binary parallel adder:
  - Total delay = (2N + 1)tp ; where tp= propagation delay of each gate.
- N bit Look ahead Addrer:
  - Total delay = 4tp ; where tp= propagation delay of each gate.
- If the n-bit decoded information has unused or don't care combinations, the decoded output will have less than 2^n outputs.
  - The decoders is also called n-to-m line decoder where $m \leq 2^n$
- Sequential Circuits:

$$\frac{1}{Clock\ Period}$$

- With N-flip flops,
  - Ring counter can have N modulus counter
  - Johnson counter can have 2N modulus counter
- Integer representation:
  - Signed Magnitude representation:
    - $-(2^{k-1}-1)\ to\ +(2^{k-1}-1)$
  - 1's complement representation:
    - $-(2^{k-1}-1)\ to\ +(2^{k-1}-1)$
  - 2's complement representation:
    - $-(2^{k-1})\ to\ +(2^{k-1}-1)$
- Duty cycle: A duty cycle is the percentage of one-period in which a signal is active.
  - So duty cycle, $D = (\frac{T}{P} \times 100)\%$
- $Time\ Period = \frac{1}{frequency}$

SOFTWARE ENGINEERING

- Risk estimation method:
  - Priority of each risk can be computed as:
    - $\boldsymbol{P = R \times S}$; R= the likelihood of a risk coming true; S = the consequences of the problems associated with that risk.
    - Basic COCOMO model:
      - Efforts in person-months, $E = a(KLOC)^b$
      - Development time in months in person-months, $D = c(E)^d$
      - where a,b,c,d are coefficients that have fixed values for different classes of projects.

COCOMO model coefficients:

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

- Cyclomatic complexity V(G) for a flow graph G with V vertices and E edges is defined as,
  - V(G) = E – V + 2
- If software project cost is K and peak development time = td then,
  - Peak manning time, $m_0 = \frac{K}{td \times \sqrt{e}}$
- If team size =n, then no. of communication channels = $\frac{n(n-1)}{2}$
- Node X – dominates node Y iff all the paths from initial node to node Y passes through node X.
- $Productivity = \frac{LOC}{(Number\ of\ programmers) \times (number\ of\ days)}$
- Cyclomatic Testing comes under white box testing.
- $Availability = (\frac{MTTF}{MTTF+MTTR} \times 100)\%$
- Span of control = number of immediate sub-routines.
- Boundary value analysis and equivalence class partitioning both are test case design techniques in black box testing.
- For a function of n variables,
  - Boundary value analysis yields (4n + 1) test cases.
- $Annual\ Change\ Traffic, ACT = \frac{(KLOC)_{added} + (KLOC)_{deleted}}{(KLOC)_{TOTAL}}$

- If T=life time of a software and initial development cost = x, then
  - Total cost of a project = Development cost + T (Development cost $\times ACT$)
- Let us assume a software tested by error seeding technique.
  - Let N be the total number of defects in the system.
  - N be the number of these defects found by testing.
  - S be the total number of seeded defects.
  - S be the number of seeded defects found during testing
  - Then, the number of defects still remaining after testing = $\frac{n(S-s)}{s}$