**1.OPERATING SYSTEM:-**

1.TLB\_hit\_time := TLB\_search\_time + memory\_access\_time

2.TLB\_miss\_time := TLB\_search\_time + memory\_access\_time + memory\_access\_time

3.Hit ratio = Total number of Hit Counts / Total number of Reference Counts

4.Effective access time = hit ratio \* time during hit + miss ratio \* time during miss

5.Effective Memory Access Time = p \* (page fault service time) + (1 - p) \* (Memory access time)

P = page fault rate = probability of page fault occur

1 – P = probability of no page fault

/\* Let P be the page fault rate OR probability that the page fault occurs \*/

6.MTTBF, the mean time between failures, is MTBF = MTTF + MTTR

7. Condition for dead lock can not occur P(N-1) + 1 <= R

P = number of processes ; n = max requirement of each process ; R = Total number of available resources

**2.SOFTWARE ENGINEERING :-**

Cyclomatic complexity = E - N + P

V(G) = E - N + 2

V (G) = P + 1

Where P = No of connected component or No of Nodes that have exit points.

N = Nodes and E = Edges

Current Failure Intensity =

Initial Failure intensity X [ 1 – Experienced failures / Failures in infinite time ]

The Basic COCOMO:-

Effort (E) = a\*(KLOC)b Man Months

Scheduled Time (D) = c\*(E)d Months(M)

Where,

E = Total effort required for the project in Man-Months (MM).

D = Total time required for project development in Months (M).

KLOC = the size of the code for the project in Kilo lines of code.

a, b, c, d = The constant parameters for a software project.

Software Maturity Index:-

The Software Maturity Index (SMI) is defined as SMI = [Mf – (Fa + Fc + Fd)] / Mf

Mf = the number of modules in the current release.

Fa = the number of modules in the current release that have been added.

Fc = the number of modules in the current release that have been changed.

Fd = the number of modules in the current release that have been deleted.

**3.FIRST PAPER:-**

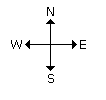


CONTRADICTORY:- Both can not be true and both can not be false.

CONTRARY:- Both can be false. Both can not be true.

SUB CONTRARY:- Both can be true. Both can not be false.

IMPLICATION:- if A is true and I is false case only not possible.



DATA INTERPRETATION:-

1. A is what percentage of B ? A / B \* 100

2. B is what percentage of A ? B / A \* 100

3. A is what percentage more than B? A – B / B \* 100

4. B is what percentage less than A? A – B / A \* 100

**4.COMPUTER ARCHITECHTURE:-**

Amdahl's Law:- Speedup(N) = 1/((1-p)+p/N);

where p= portion of the code that can be made parallel

N=number of processor. For example:

N = The part which performance needs to be improved.

Execution time after improvement = Execution time unaffected + Execution time affected / Amount of improvement;

Total time required to complete n tasks in k segment pipeline with tp clock cycle time:

= ( k + n − 1 )tp

Speedup ratio = time taken without pipeline / time with pipeline

**5.COMPUTER NETWORK:-**

**Hierarchical Routing:-**

Minimum size of Routing table = (Cluster – 1) + (Region – 1) + Router

Cluster \* Region \* Router = Total Router

**Shannon-Hartley Channel Capacity Theorem :-**



**S / N** isSignal to noise ratio, B is Bandwidth.

**Nyquist Criteria for maximum data rate for noiseless channels**

C = 2 \* B \* log M

where C is the channel capacity in bits per second or data rate,

B is the maximum bandwidth allowed by the channel,

M is the number of different signaling values or symbols

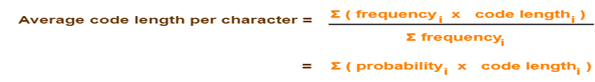
log is to the base 2

**Hamming Distance**

The minimum Hamming distance is 2t + 1, the code can correct up to t errors.

For error detection, formula is t+1.

**Huffman Code or Optimal Coding Technique**



**RSA ALGORITHM:-**

\*) n = p \* q; m = p-1 \* q-1; GCD(e, ((p-1) \* (q-1))) = 1 ;

\*) Encrypt Message E(s) = se mod n

\*) Find d, such that de % m = 1

\*) Decrypt Message E(s) = encryptedd mod n

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

Transmission rate = frame rate \* number of bits in a slot

Maximum burst time = Capacity / ( Output rate-Arrival rate )

Total delay = routes \* packets / transmission rate.

**6. DATA STRUCTURES AND ALGORITHM**

Time Complexity of Topological Sort O(V+E)

**Greedy Algorithm:-**

-Prim's Algorithm O(E log V)

-Kruskal Algorithm O(E log V)

- Dijkstra Single Source Shortest Path Algorithm O(V2)

-Fractional Knapsack problem

-Huffman encoding Alg

**Dynamic Programming**

-Floyd Warshall Algorithm All pair shortest path (Time Complexity: O(V^3) )

-Bellman-Ford algorithm Single Source Shortest path O(V + VE + E) = O(VE).

**Radix Sort Time complexity:-**

Let there be d digits in input integers. b is the base for representing numbers, for example, for decimal system, b is 10.the set array {1, 2, ..., n}

Radix Sort takes O(d\*(n+b)) time

Total Number of Binary trees with n nodes is 

**7. DATABASE MANAGEMENT SYSTEM:-**

Check for the highest Normal Form:-

2nd Normal Form:- Defn1:- LHS must be proper subset of any candidate key and RHS must be a non prime attribute

Defn 2:- It should not be partial dependency. Partial dependency means subset of candidate key is determining non prime attribute.

3rd Normal Form:- Either LHS is CK or RHS should be prime attribute. Otherwise it is not in 3rd NF.

BCNF:- LHS of all FD should be CK or SK

BTREE Indexing:-

Size of non leaf nodes = m \* (block pointer) + (m-1) \* (key + record pointer);

Size of non leaf nodes = m \* (block pointer) + (m-1) \* (key) + (m-1) \* (record pointer);

m \* (block pointer) + (m-1) \* (key) + (m-1) \* (record pointer) <= BlockSize

**8.AUTOMATA & COMPILER DESIGN**

