Computer Science & IT Engineering

OS & Algo

By- Dr. Khaleel Sir

Important Terms & Formulas

Operating Systems

1. Introduction & Background

N. Pr [Lack of resp.]

Pr [

a) U. Pr vs M. Pr

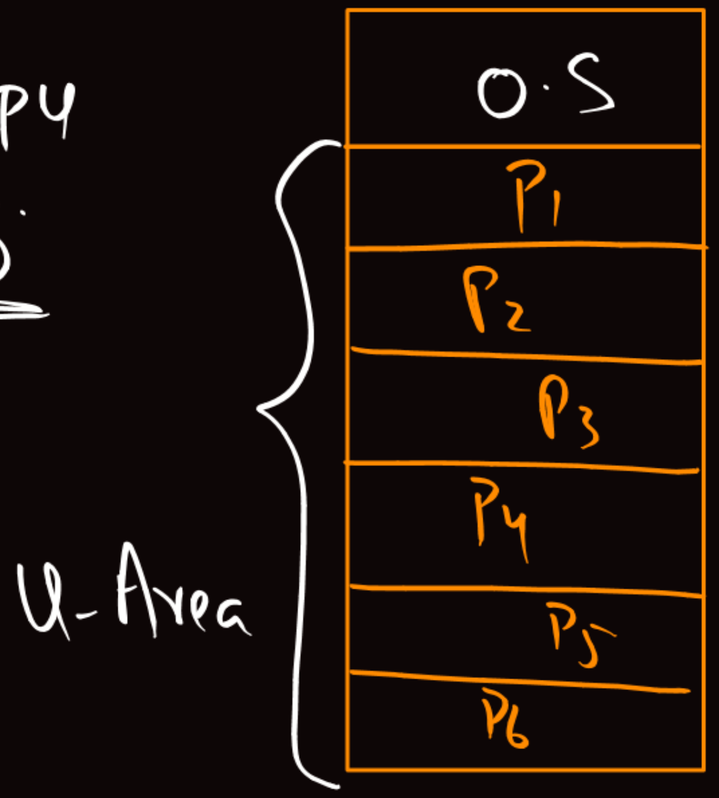
→ low cpu utiliz.

Memory (Max. cpu utiliz.

↓

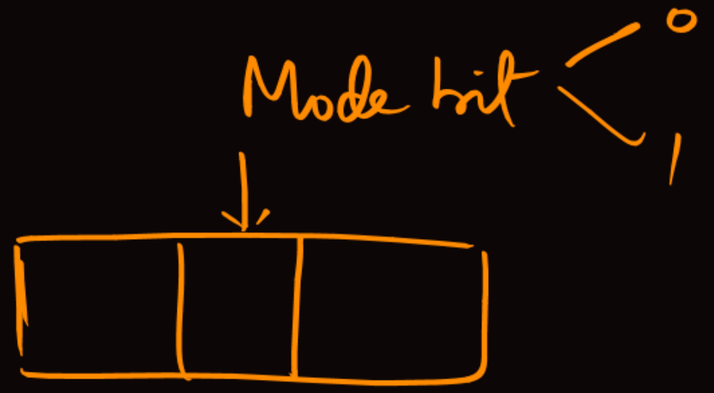
Throughput

= No. of Programs/Process / Unit Time



⇒ Dual Mode Support from CPU

- **User Mode** — Non-Atomic
- **Kernel Mode** — Atomic



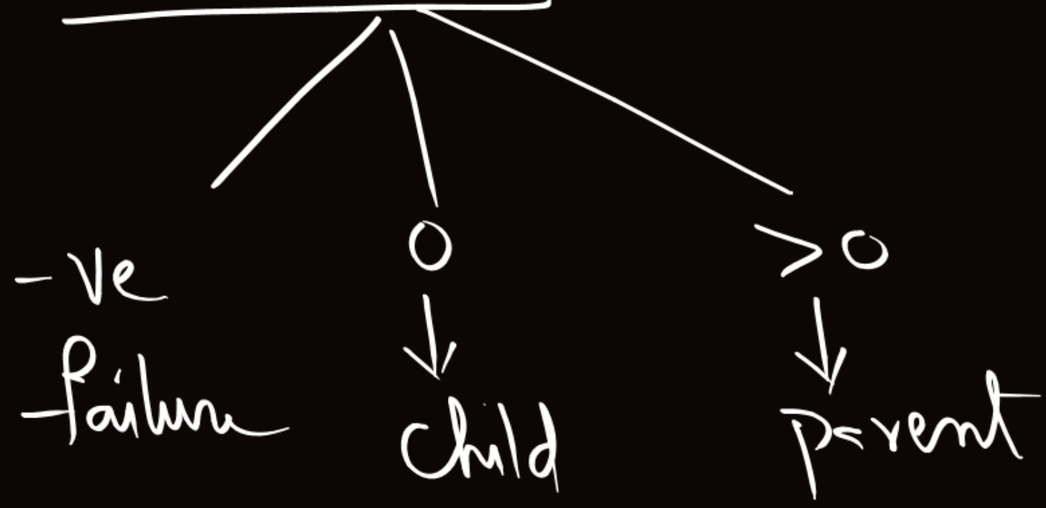
Mode Shifting:

→ System Call [fork()]

fork(): create a child/New Process

	Total Processes	New child
① n-fork() in Series :	2^n	$(2^n - 1)$

② Return-value :



(ii) exec : execute/Run an application in the Process

(iii) setpid() :

(iv) wait() :

→ All user defined fns
→ All Library fns
} U-M

→ O.S Routines / Calls } K.M

Process Management

(i) Process Concepts

(ii) Process Structure

- Code Section/Text
- Data "
- Stack
- Heap : DMA
  - malloc() | New

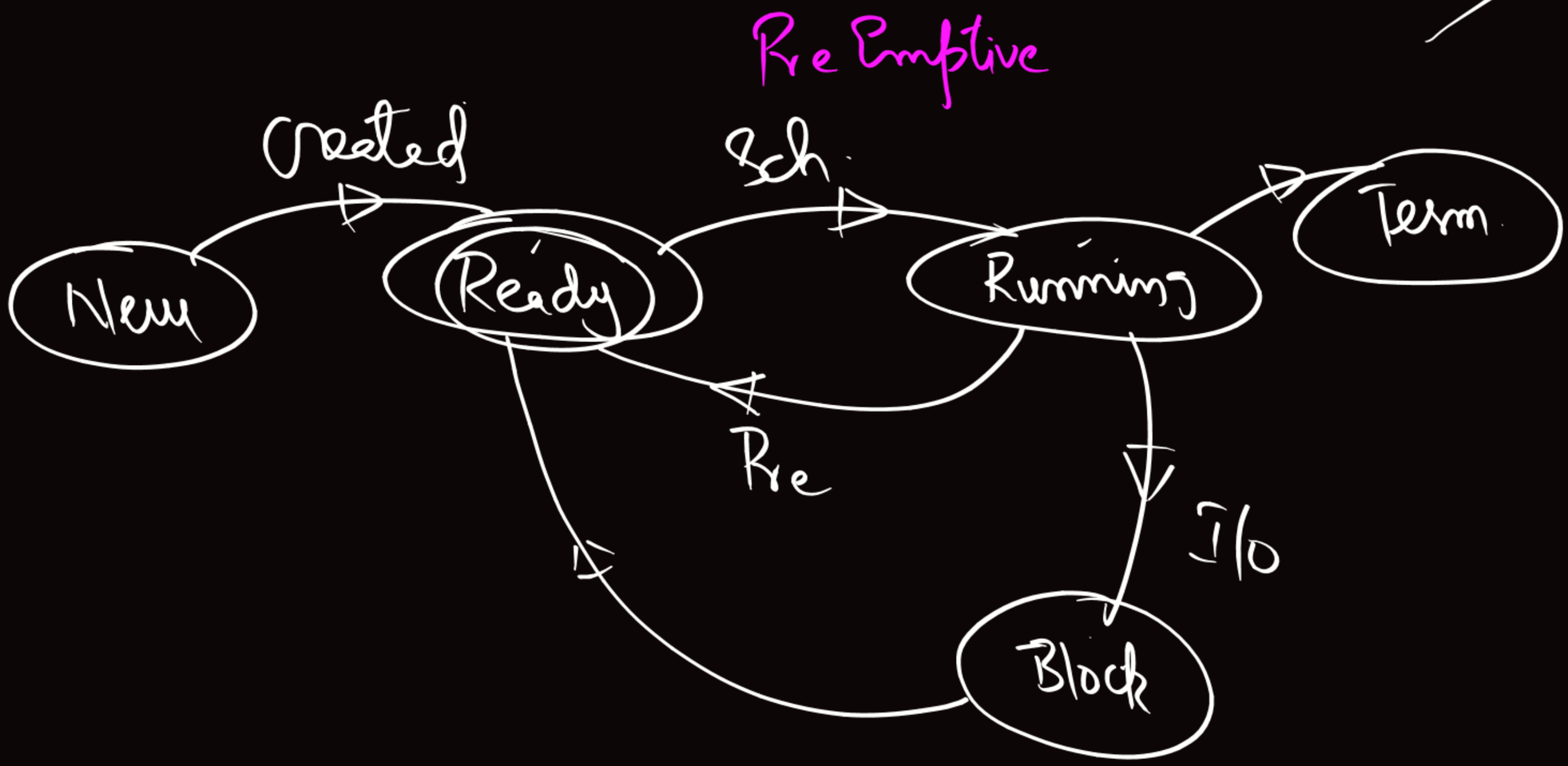
→ as an A.D.T

- → operations
- → Attributes: State

P.C.B / Process Descriptor } Process Content

States & Transition Diagram

<New; Ready; Running; Block; Terminate>



Schedulers & Dispatcher → Context switching

Deg. of M.Pr

L.T.S : New → Ready

S.T.S : Ready → Running

M.T.S : Process Suspension & Resumption

↓

Dispatch Latency

CPU Scheduling:

(i) Process Times:

a) Arrival Time (AT):

b) Burst Time (CPU BT):

c) IO Burst Time (IOBT)

d) Completion Time (CT)

e) Waiting Time (WT): $\bar{TAT} - (\bar{BT} + \bar{IOBT})$ If $\bar{IOBT} = 0$, $\bar{WT} = \bar{TAT} - \bar{BT}$

f) Turn Around Time ($\bar{TAT}$): $\bar{CT} - \bar{AT}$

For a given Set of n-processes

$$\text{Schedule length}(L) = \text{Max}(CT_i) - \text{Min}(AT_i)$$

$$\text{Throughput} = n/L$$

n — Units

? — 1.

Response Time:

of Process Request

↳ The time q Req. admission to the time @ which it generates its first result

Process

: The difference q the time q arrival to the time @ which the Process gets onto CPU for first time;

(Non-PreEmptive + No I/OBT)

(i) FCFS

(ii) SJF

(iii) SRTF

(iv) HRRN

(v) Round Robin

(vi) Priority

(vii) R.R + Priority

(viii) Multi-Level Q-Scheduling

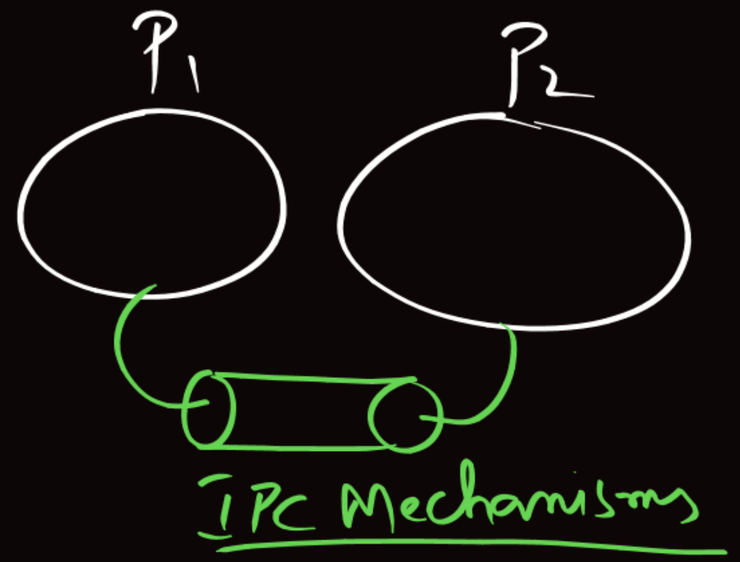
Threads (MSQ's)

→ Process vs Threads

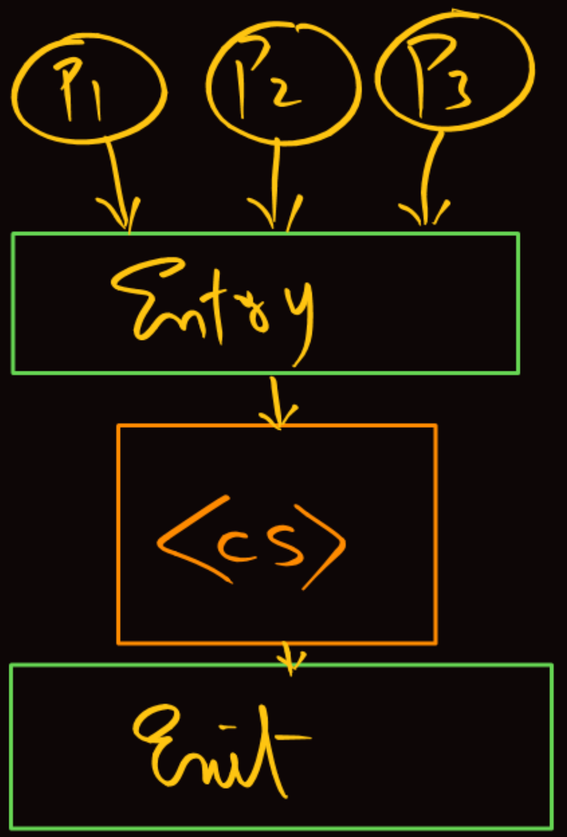
→ Benefits of Threads

→ Types of Threads

- U-L-Ts
- K-L-Ts

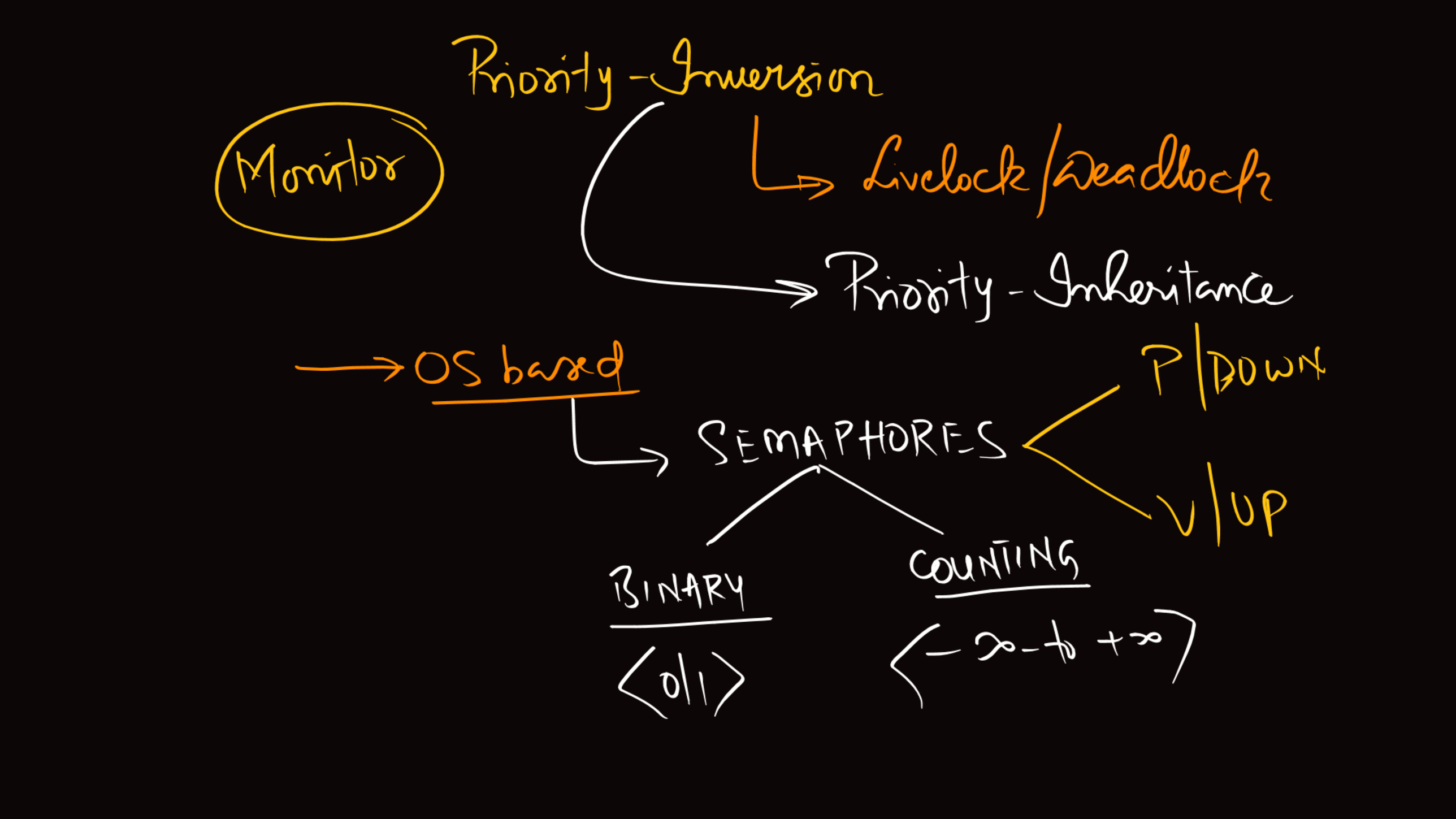


Process Synchronization/Coordination



- Critical Sec
- Race Conditions
- PreEmption

(i) Mutual Exclusion
(ii) Progress
(iii) Bounded wait
- (Starvation)
- Fair-Share


Priority - Inversion
Monitor
Livelock / Deadlock
Priority - Inheritance
OS based
SEMAPHORES
P / DOWN
V / UP
BINARY
<0/1>
COUNTING
<-∞ - 0 +∞>

Concurrency Mechanisms

- fork - join
- Parbegin - Parend

Deadlocks:

- Concept
- Sys. Model
- Nec. Conditions
- R.A.G

Strategies

- Proactive
  - Prevention:
    - Avoidance (Bankers)
      - Safe
      - unsafe
  - Ignorance: (Ostrich)
- Reactive
  - Detection & Recovery

(III) Memory Subsystem :

n = Addr. bits

N = No. of words

m = word length in bits

$$N = 2^n \text{ words} \quad (n\text{-bits})$$

$$n = \lceil \log_2 N \rceil \text{ bits}$$

$N = 16\ GB$

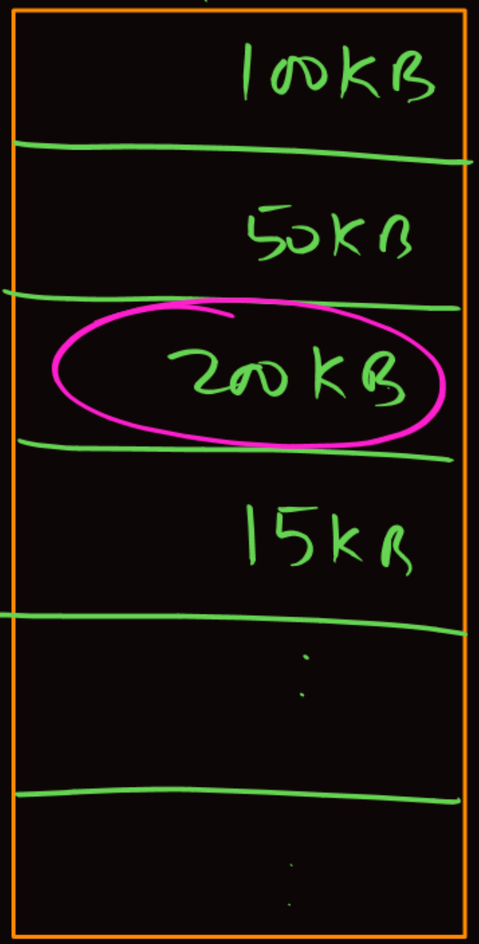
$n = 30 + 4 = 34$ bits

$$N = \log_2 \times \text{Bytes}$$

$$n = \log \log \times \text{bits}$$

M. Mgmnt Techniques

- CG
  - → Partitions
    - Fixed (MFT)
    - Variable (MVT)
  - → overlays
- Non-CG



Part. Alloc. Policy
- FF
- BF
- W.F
- N.F

Protection
- Limit & Base Registers

(i) Fragmentation:
- Internal ✓
- External ✗

Variable Partitions :

→ No Int. Frag.

→ Ext. Frag:
- Compaction
- N.C.G

Paging:

$$N_{pages} = \frac{V.A.S}{P.S}$$

$N = 2^P \Longleftarrow P = \log_2 N_{pages}$ bits

$P.S = 2^d$ [d = Page offset]

$d = \log_2 PS$

VA / LA : | P | d |

P.T. Size = N * e Bytes

N = No. of Pages

e = P.T.E Size in Bytes

Paging with TLB

$$EMAT_{SP+TLB} = \alpha(c+m) + (1-\alpha)(c+2m)$$

Virtual Memory (VM)

→ Demand Paging (DP) < Pure, Prefetch.

$$EMAT_{DP} = (1-P)\,m + P * S$$

Galvin

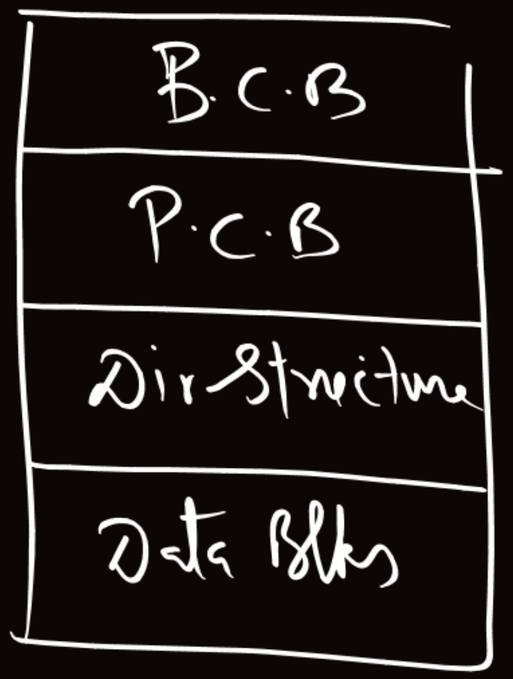
⟹ Page replacement < LRU, optimal, FIFO | LIFO

⟹ Thrashing

⟹ Working Set Model

File System & IO Mgmnt

- Physical Disk Structure
- Logical " "



DBA :

DBS :

$2^{DBA} * DBS$

Max. Possible File Size : Max. Possible Disk Size

(i) Alloc. Methods

- C.S :
- Linked :
- Index :
  - I-Node Structure*

F.A.T

Disk Scheduling

(i) Seek Time

< FCFS ; S.S.T-F ; SCAN ; LOOK

C SCAN ; CLOOK >

ALGORITHMS

1. Asymptotic Notations [DS + A.S.N]

→ Log's + Series + Exponentials + Combinatorics + Prob + Diff + Integration

Defns

- Big
  - O, Ω
  - Θ
- Small
  - o, ω

Properties of ASN

Fns

II. Divide & Conquer :

(i) MaxMin

(ii) Merge Sort

(iii) Quicksort

(iv) Matrix Multip.

(v) Binary Search

(vi) Power of n

(vii) L.I.M

Recurrence Equations

(i) MaxMin : $T(n) = 2T(n/2) + 2 \Rightarrow O(n)$ → $\frac{3n}{2} - 2$

(ii) Binary Search : $T(n) = T(n/2) + c \Rightarrow O(\log n)$

(iii) Merge Sort : $T(n) = 2T(n/2) + n \Rightarrow O(n \log n)$

(iv) Quick Sort :
- W.C : $T(n) = T(n-1) + n \Rightarrow O(n^2)$
- B.C : $T(n) = 2T(n/2) + n \Rightarrow O(n \log n)$

(v) Mat. Mult : $T(n) = 8T(n/2) + n^2 : O(n^3)$

$= 7T(n/2) + n^2 : O(n^{2.81})$

Master Method

$$T(n) = a \cdot T(n/b) + f(n)$$

Case I :

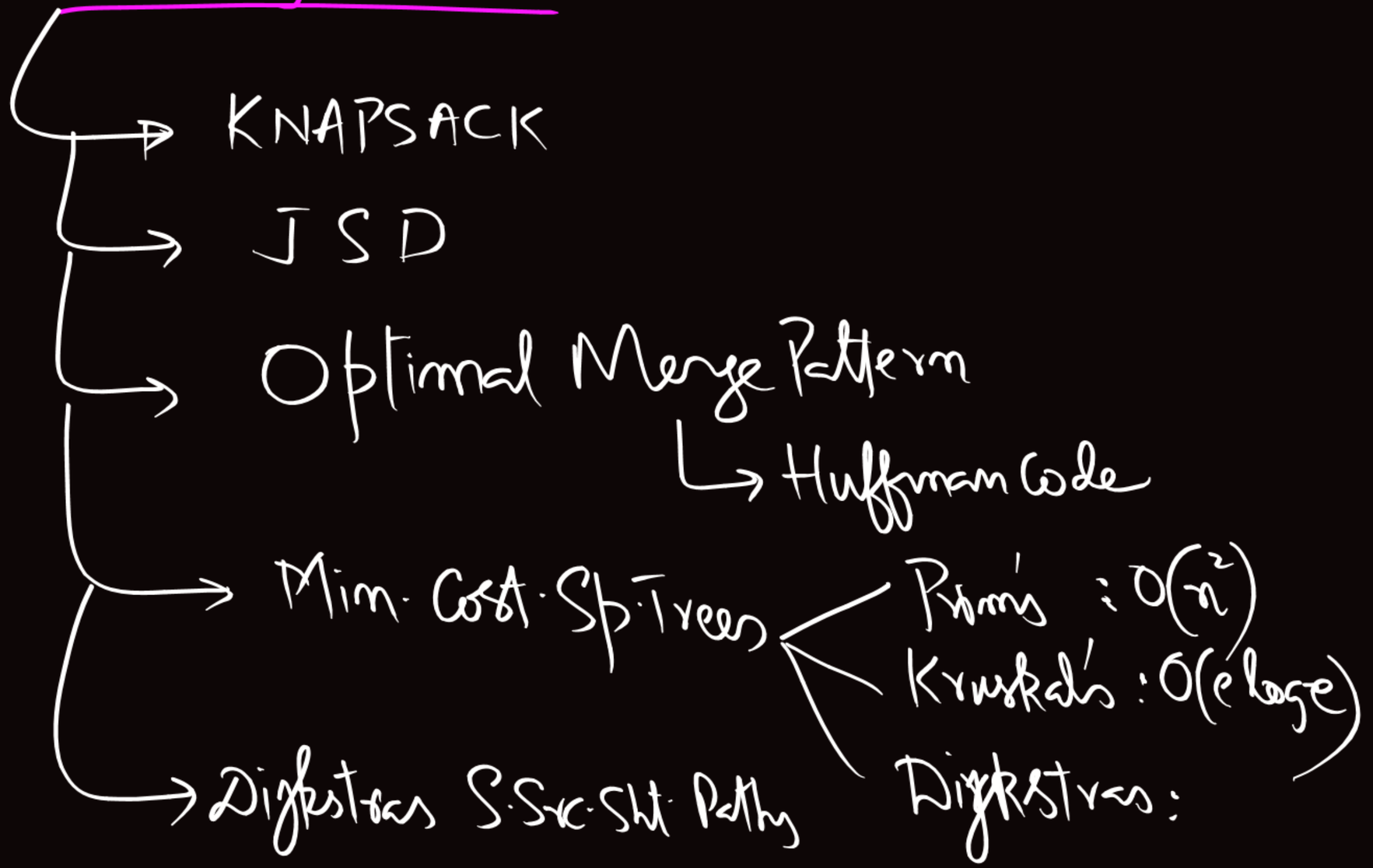
Case II :

Case III :

L.I.M :

1) DC : $T(n) = 4T(n/2) + n$
2) AK : $T(n) = 3T(n/2) + n$
3) TC : $T(n) = 5 \cdot T(n/3) + n$

Greedy Method



Dynamic Programming:

- opt. Substructure Property
- overlapping

- Fib
- M.S. Graphs
- T.S. Person
- * Longest Common Subsequence
- * Matrix-chain Product
- * Sum of Subsets
- opt. Binary Search Tree

(i) Soln (D. Pros. based Recurrence)

(ii) Numerical

(iii) Time & Space

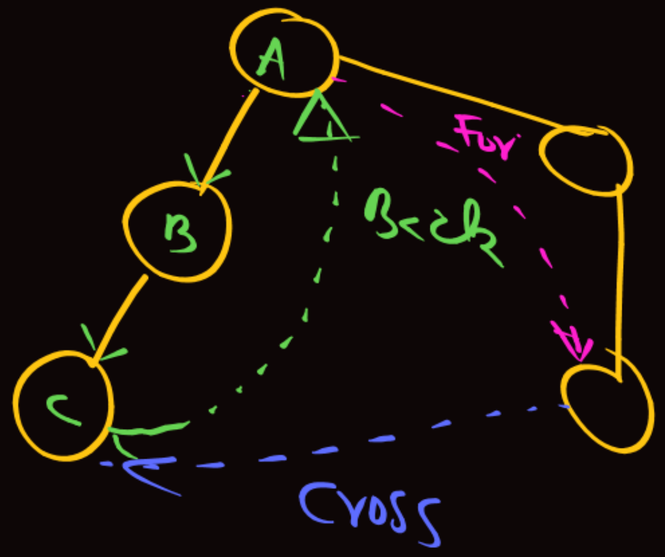
Graph Techniques

- B.F.S + D.F.S
- Applications
  - Components
  - Topological Sort *

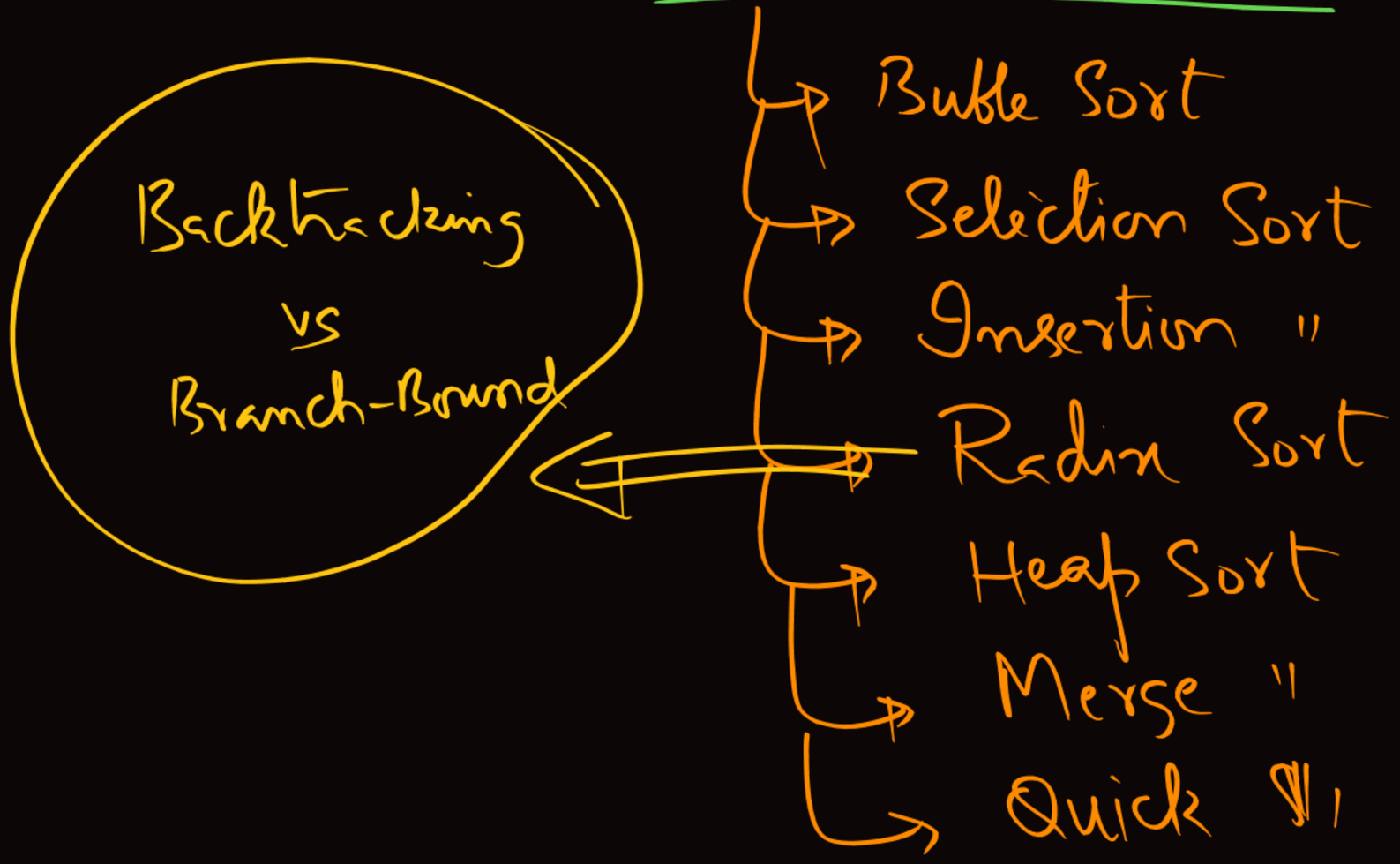
Tree edge
Forward "
Back "
Cross "

→ Heap Algo's:

- Heap Sort



SORTING METHODS



1) Best Case, Worst Case } Time
2) Space Complexity
3) Stable or not
4) Inplace or out of Place