CS & IT ENGINEERING

Algorithms

Miscellaneous Topics

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic Divide and Conquer

Topics to be Covered

Topic

Introduction to Greedy Method

Control Abstraction

Knapsack Problem

GREEDY METHOD:

→ Used for Solving Problems, whose Solutions are viewed as a result of making a Set/Sequence of Decisions;

→ These Decisions are made in a Step-wise manner;

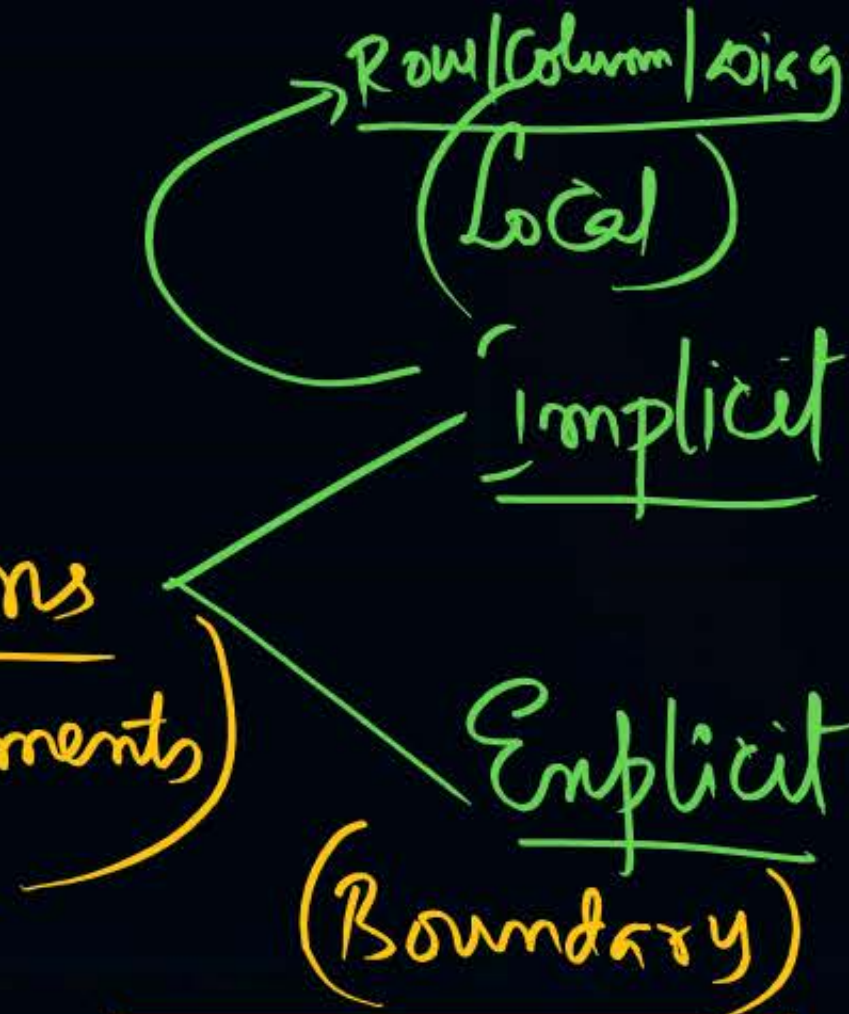→ At each Step out of all options, Greedily Select that option, which Satisfies the given criteria of the problem;

Terminology:

→ Problem Definition,

→ Constraints (Conditions / Requirements)

→ Solution Space :

" All possible ways of organizing inputs, Satisfying only explicit Constraints "

Row | Column | Diag

(Local)

implicit

Explicit (Boundary)

$$1 \leq x[i] \leq n$$

$\langle 1, 2, 3, 4 \rangle^{*}$

① n-Queens :

n-queens $(q_1 \dots q_n)$

n = 4 ; $(q_1 \dots q_4)$

	1	2	3	4	
q_1		•			x[1]
q_2	x	x	x	•	x[2]
q_3	•				x[3]
q_4	x	x	•		x[4]

int $x[1 \dots n]$, $x : \langle 2, 4, 1, 3 \rangle$
$x : \langle 3, 1, 4, 2 \rangle$

$x[i] = pos$ (Column) in which Q_i is placed

$n = 4$

$X \begin{bmatrix} 1, 2, 3, 4 \end{bmatrix} \checkmark$

$\begin{bmatrix} 1, 2, 4, 3 \end{bmatrix}$

$\begin{bmatrix} 1, 3, 2, 4 \end{bmatrix}$

$\begin{bmatrix} 2, 4, 1, 3 \end{bmatrix}$

:
:

$n!$
$\underline{\underline{}}$

$n = 3$

1, 2, 3
1, 3, 2
2, 1, 3
2, 3, 1
3, 1, 2
3, 2, 1

$\rightarrow$ Each problem will have its Solution space as a fn' of Input Size

Eg: for $n-Q's$ Problem, it is $n!$
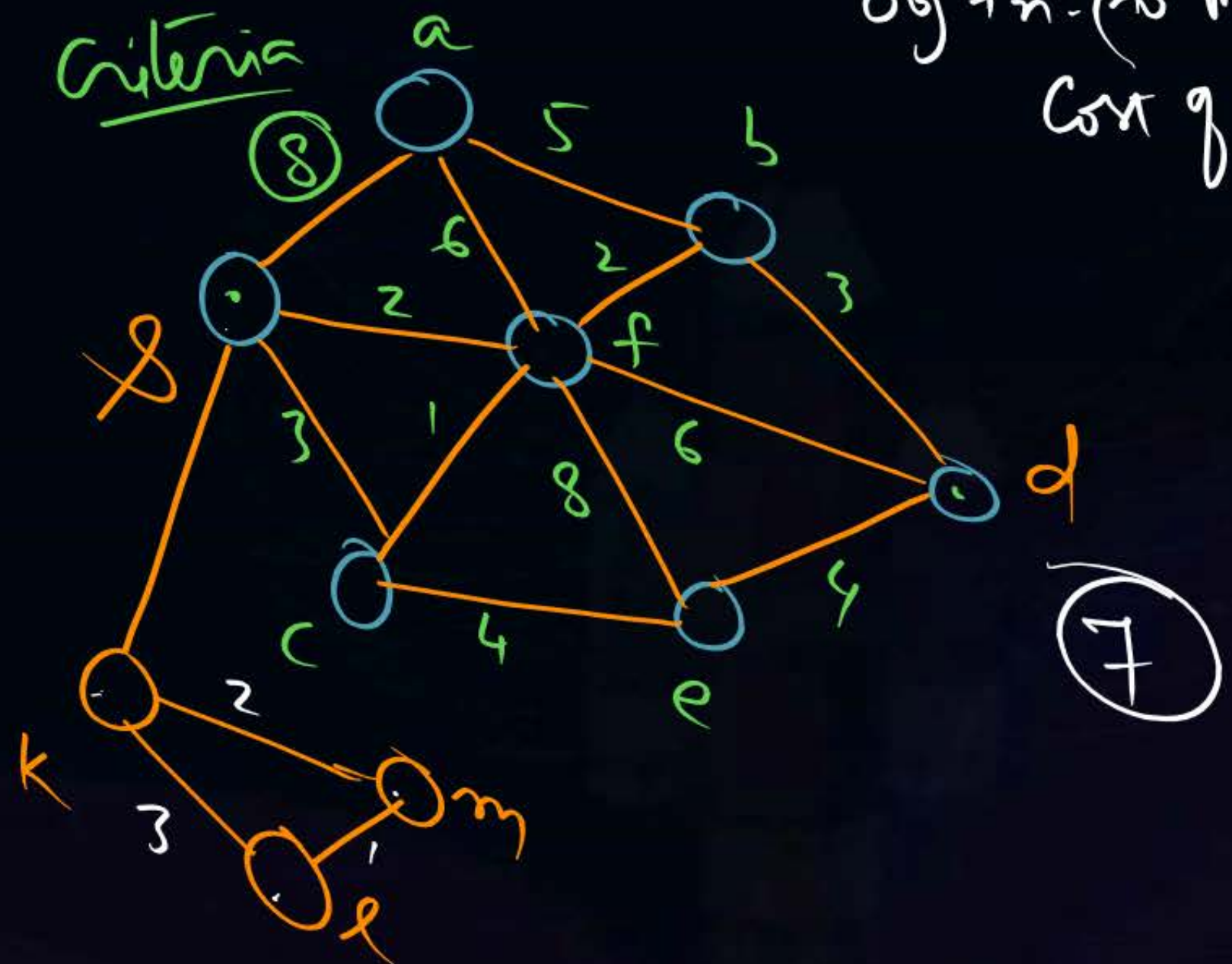
: 2^n ; ∞ ; n^2 ; n^n

→ **Feasible Solutions**: are those Solutions in the Solution Space that Satisfy implicit Constraints of the Problem, Problems may have multiple Feasible Solutions,

→ **Objective function:**
(Few problems may have Objective fn, which tries to Min./Max. a given Criterion of the Problem)

→ **Optimal Solution:** is that feasible Solution, which satisfies the objective fn;

Ex: **Shortest Path**

obj. fn: (to minimize Cost of Path)

Criteria a

⑧

8

5   b

6   2   3

8   2   f

3   1   6

8   6

c   4   e   4   d

k   2

3   m

l   1

⑦

$\rightarrow$ Optimal Solution always refers to the value and hence it is always unique;

Q) What is objective fn of N-Queens Problem?

$\rightarrow$ No objective fn

Feasible Soln

Searching

n-queens

Sorting

Problem (P)

(Shortest Paths)

Graph Coloring

Decision

→ Its result is
always either Y/N
(T/F) or $(0/1)$

(Feasible Soln)

Optimization

→ Requires to
determine a Max/Min
values of a given
criterion;

(optimal Soln (objective)
Fn

1) CPU Scheduling $\longrightarrow$ Optimization

2) Page replacement $\longrightarrow$ Optimization

3) Disk Scheduling $\longrightarrow$ Optimization

4) Banker's Safety Algo $\longrightarrow$ Decision

5) GATE 2024 Problem

6) Congestion Control $\longrightarrow$ Optimization

$$\frac{G \ CSJT}{G \ DA}$$

Topic : Greedy Method

Control Abstraction
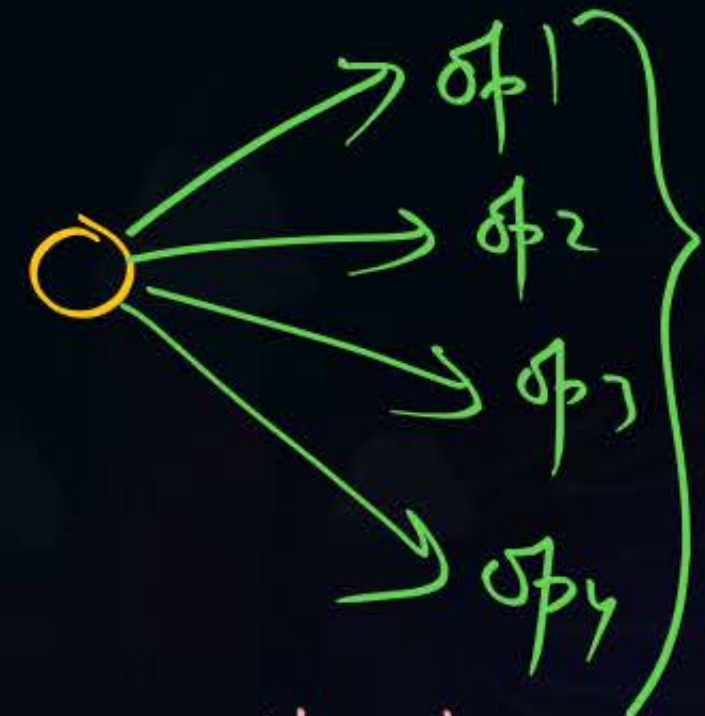General Method

1. Algorithm Greedy(a, n)
2.   // a[1 : n] contains the n inputs.
3.   {
4.   1. solution := $\emptyset$; // Initialize the solution.
5.   2. for i := 1 to n do
6.   { $\rightarrow O(1)$
7.   x := Select(a);
8.   if Feasible(solution, x) then
9.   solution := Union(solution, x);
10.   } $\rightarrow O(\log n)$ ADD
11.   return solution ;
12.   }

$O(n)$ $O(1)$

(Principle of Local Optimality)

$\rightarrow op1$
$\rightarrow op2$
$\rightarrow op3$
$\rightarrow op4$

Time complexity of atleast $O(n)$

Slide 4

1. KNAPSACK Problem
(KNAP)

→ Given a KNAPSACK of Capacity M
  (Bag)

→ Given n - objects $(O_1, O_2 \cdots O_n)$

- ✓ weight (W_i)
- Profit (P_i) — (Max. Profit)
- Decision (x_i) ?
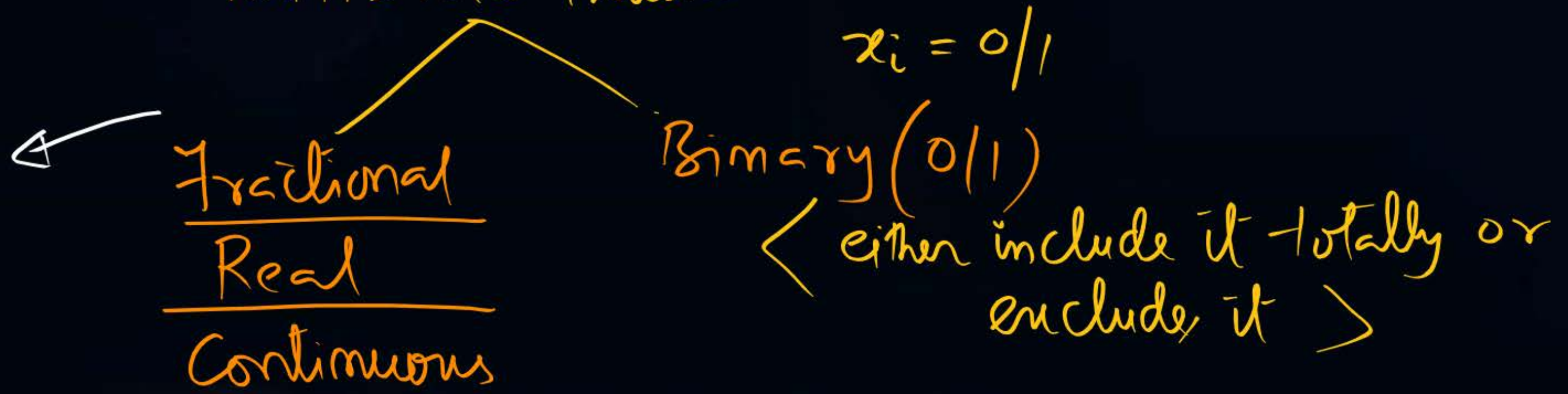
→ If object of weight (W_i) is put into the KNAP, then KNAP gets filled up by wt. W_i & we get profit P_i

→ Maximize the Profit Subject to the Condition that the total wt. being put into the KNAP. does not exceed its Capacity.

KNAPSACK Problem

$$x_i = 0/1$$

Fractional
___
Real
___
Continuous

Binary $(0/1)$

$\langle$ either include it totally or exclude it $\rangle$

$$\boxed{0 \leq x_i \leq 1}$$

$\langle \quad \infty \quad \rangle$

1) $\sum\limits_{i=1}^{n} w_i \leq M \implies$ Total Profit $= \sum\limits_{i=1}^{n} P_i$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \hookrightarrow x_i = 1$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \langle \text{No-Decision} \rangle$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \langle \text{Trivial} \rangle$

2) For Solving the Problem, then

$$\sum\limits_{i=1}^{n} w_i > M$$

Explicit Constraint

$\left(\begin{array}{c} \text{Decision} \\ \text{on } O_i \end{array} \right) x_i \left\langle \begin{array}{c} w_i * x_i \\ \\ P_i * x_i \end{array} \right.$

$x_1 = ?$

$x_2 = ?$

$x_3 = ?$

$$\text{Max} \sum_{i=1}^{n} P_i * x_i \longrightarrow \text{object. } \text{Fn}$$

$$\text{Subject to} \sum_{i=1}^{n} W_i * x_i \leq M \longrightarrow \text{Impl. constn't}$$

$$\text{Where} \quad 0 \leq x_i \leq 1 \Longrightarrow \text{Expl. constraint}$$

$$\langle x_1, x_2 \cdots x_n \rangle$$

Soln Space

∞

(G.M)   (D.P)

Fractional   0,1 KNAP

"∞"   "2^n"

$$x_i = 0|1$$

$$\langle x_1, x_2 \cdots x_n \rangle$$

2^n

① $m = 3;$ $M = 20;$ $\langle w_1; w_2; w_3 \rangle = \langle 18; 15; 10 \rangle$

$\langle P_1; P_2; P_3 \rangle = \langle 25, 24, 15 \rangle$ $\underset{0_1}{=}$

(Fractional KNAP)

$\langle x_1 \; x_2 \; x_3 \rangle$

a) Greedy about Profit:

$x_1 = 1$
$x_2 = 2/15$
$x_3 = 0$

$\sum w_i x_i = 20$

$\sum P_i x_i = \left(25 + \dfrac{2}{15} \times 24 + 0 \right)$

$= \boxed{28.3}$

b) Greedy about weight:

$x_3 = 1$
$x_2 = 10/15 = 2/3$
$x_1 = 0$

$\sum w_i x_i = 20$

$\boxed{\sum P_i x_i = 31}$

$w_i \longrightarrow P_i$

$1 \rightarrow ?$ $\left(P_i / w_i \right)$

1) $T(n) = 2T(\sqrt{n}) + \dfrac{\log n}{\log \log n}$, $n > 4$

$ = c$, $n \leq 4$

2) $T(n) = 2T(n/2) + \dfrac{n}{\log n}$, $n > 1$

$ = c$, $n = 1$