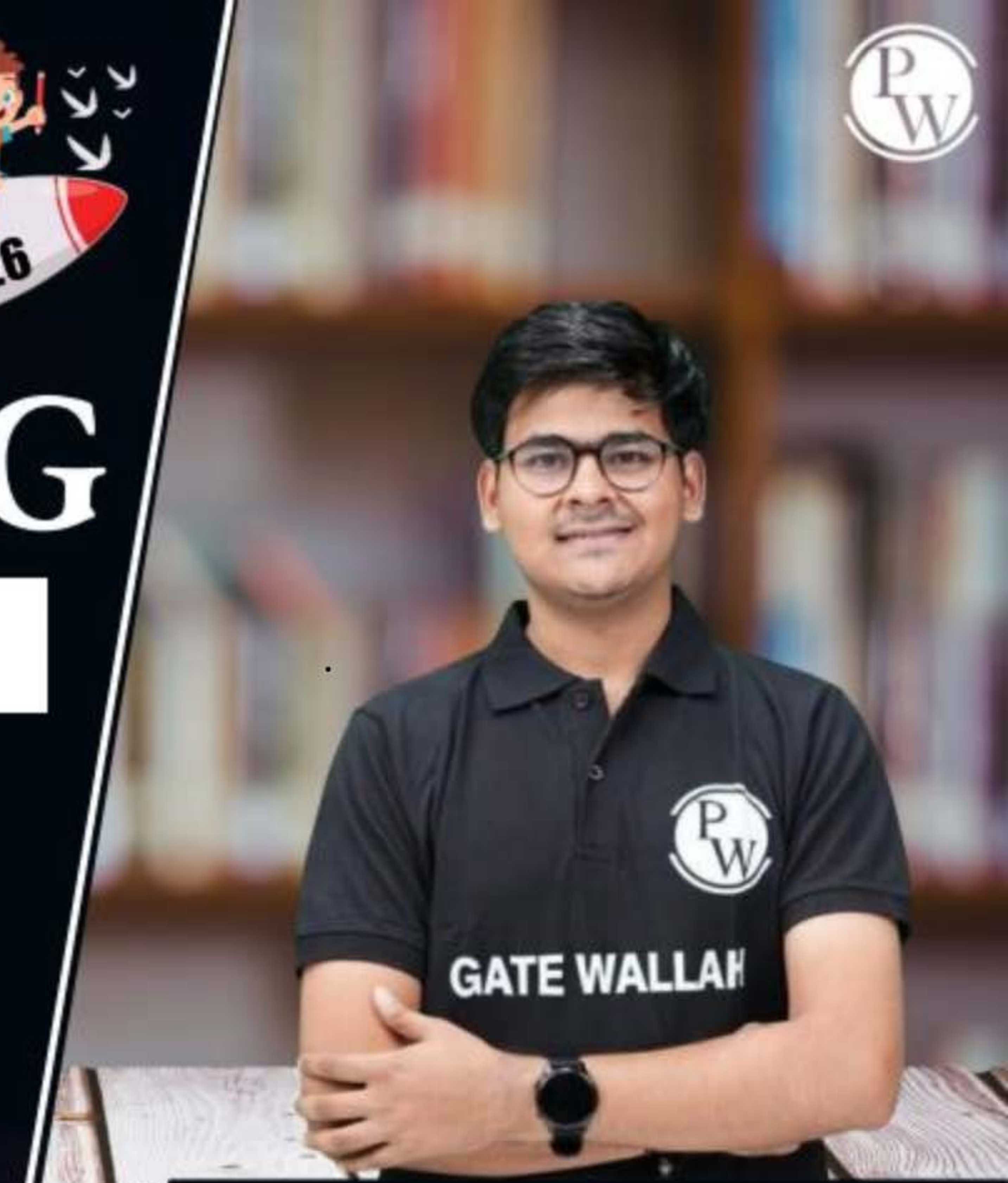
CS & IT ENGINEERING

Algorithms

Greedy Method



Lecture No.- 02

By- Aditya sir

Topics to be Covered

Topic

Topic fractional Knapsack

Topic JSD

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

@ADITYASIR_PW

Telegram Link for Aditya Jain sir: https://t.me/AdityaSir_PW

Topic : Greedy Algorithm

[NAT]

#Q. Find max/optimal solution to the fractional knapsack problem:

Given: $n = 7$, $M = 15$

Object	Profit	Weight
01	10	2
02	5	3
03	15	5
04	7	7
05	6	1
06	18	4
07	3	1

H/W1
HW2

Topic : Greedy Algorithm

Procedure GREDY_KNAPSACK(P, W, M, X, n)

// P(1 : n) and W(1 : n) contain the profites and weights respectively of the n//

// knapsack size and X(1 : n) is the solution vector//

Read P(1 : n), W(I : n), X (1 : n), M, cu;

integer i, n;

x ← 0 // initialize solution to zero//

cu ← M // cu = remaining knapsack capacity//

for i ← 1 to n do

    if W(i) > cu then exit endif

    X(i) ← 1

Assumption :- Objects are sorted in decr value of P/W.

take fraction

1 0 1-fractional

Topic : Greedy Algorithm

cu ← cu –W(i)

Repeat

if i ≤ n then X(i) ← (cu/W(i))endif

end GREEDY–KNAPSACK

2 10 → 2/10

Topic : Greedy Algorithm

PYQ

[NAT]

#Q. Consider the weights and values of items listed below. Note that there is only one unit of each item.

item number	Weight (in kg)	Value (in Rupees)
1	10	60
2	7	28
3	4	20
4	2	24

$M = 11$

$(2,3) \rightarrow 48$

$(3,4) \rightarrow 44$

$(2,4) \rightarrow 52$

$(1) \rightarrow 60$

$V_{opt} = \underline{\underline{60}}$

Topic : Greedy Algorithm

[NAT]

#Q. The task is to pick a subset of these items such that their total weight is no more than 11 kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by V_{opt}. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by V_{greedy}. The value of $V_{opt} - V_{greedy}$ is ____.

0/1
KS

Topic : Greedy Algorithm

PYQ

[NAT]

#Q. Consider the weights and values of items listed below. Note that there is only one unit of each item.

P/w

item number	Weight (in kg)	Value (in Rupees)
1	10	60
2	7	28
3	4	20
4	2	24

$x=0$

$\rightarrow 60/10 = 6$ — 2

$x=0$

$\rightarrow 28/7 = 4$ — 4

$x=1$

$\rightarrow 20/4 = 5$ — 3

$\rightarrow 24/2 = 12$ — 1

$x_i = 1$

$m = 11 - 2 = 9$ ✓

$9 - 4 = 5$ //

$V_G = 20 + 24 = 44$

$$V_0 - V_a = 60 - 44$$
$$= \underline{16}$$

Topic : Greedy Algorithm

Important Terminologies:

1) Problem Definition:

2) Solution Space:

- All possible arrangements
- Only satisfies explicit constraints.

3) ~~Problem Definition:~~ Feseable Soln:-

- Those solutions from solution space that also satisfies implicit constraints.

Constraints

Implicit ↓ (local) no walk

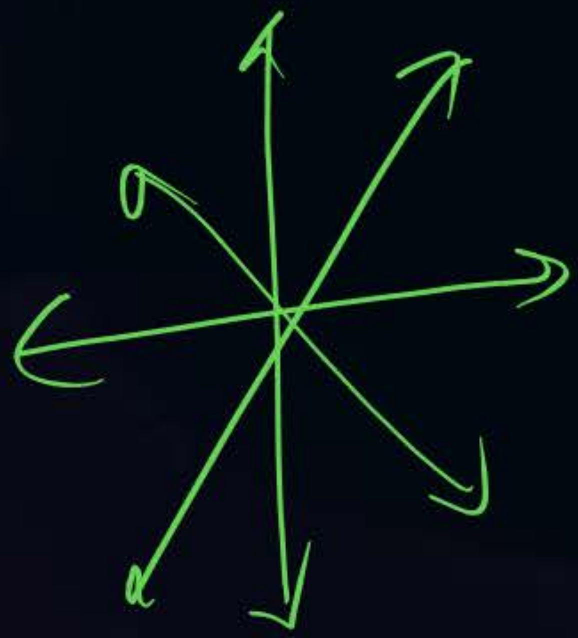
Explicit ↓ Boundary e.g.: __ ≤ × ≤ __

Example:

N-Queens Problem:

- Given n-Queens on a n × n chess board, arrange them such that no two queens should attack each other.

E.g.: $n = 4: q_1 \quad q_2 \quad q_3 \quad q_4$



	1	2	3	4
q_1				
q_2				
q_3				
q_4				

Topic : Greedy Algorithm

E.g.: $n = 4: q_1 \quad q_2 \quad q_3 \quad q_4$

$X[1 \ldots n]$

$X[i] \rightarrow$ position of q_i in the i^{th} row

$\{1 \leq X_i \leq n\}$

	1	2	3	4
q_1				
q_2				
q_3				
q_4				

Topic : Greedy Algorithm

E.g.: $n = 4$

$X = [1, 2, 3, 4]$ ✗

$X = [2, 3, 1, 4]$ ✗

$X = [4, 3, 2, 1]$

:

:

:

Size of solution space = $n!$

	1	2	3	4
q_1		0		
q_2			0	
q_3	0			
q_4				0

Topic : Greedy Algorithm

X = [1, 3, __, __]

- This is in solution space,
- But not feasible solution.

	1	2	3	4
q_1	q_1			
q_2			q_2	
q_3	×	×	×	×
q_4				

Topic : Greedy Algorithm

$X = [2, 4, 1, 3]$

	1	2	3	4
q_1		q_1		
q_2				q_2
q_3	q_3			
q_4			q_4	

$X = [3, 1, 4, 2]$

	1	2	3	4
q_1			q_1	
q_2	q_2			
q_3				q_3
q_4		q_4		

- Both are feasible solution.

$4! = 24$

Topic : Greedy Algorithm

Important Terminologies:

4) Objective function:

- Optional
- Tries to optimize (minimize/maximize) a given criteria of the problem.

Topic : Greedy Algorithm

Example:

1) Knapsack → Maximize profit
2) Shortest Path → Minimize path cost

Topic : Greedy Algorithm

Important Terminologies:

5) Optimal Solution:

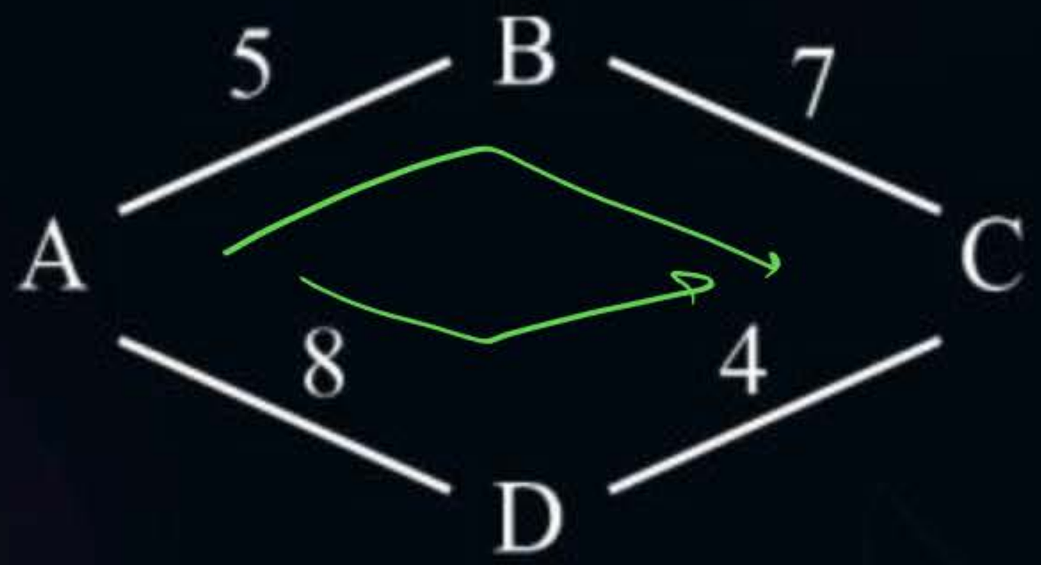
- It is that feasible solution that satisfies the objective function.
- Optimal solution always refers to the value and hence it is unique.

min/max

144

Topic : Greedy Algorithm

Example:



- Min cost A → C : 12 (unique)
- Shortest Path A → C :

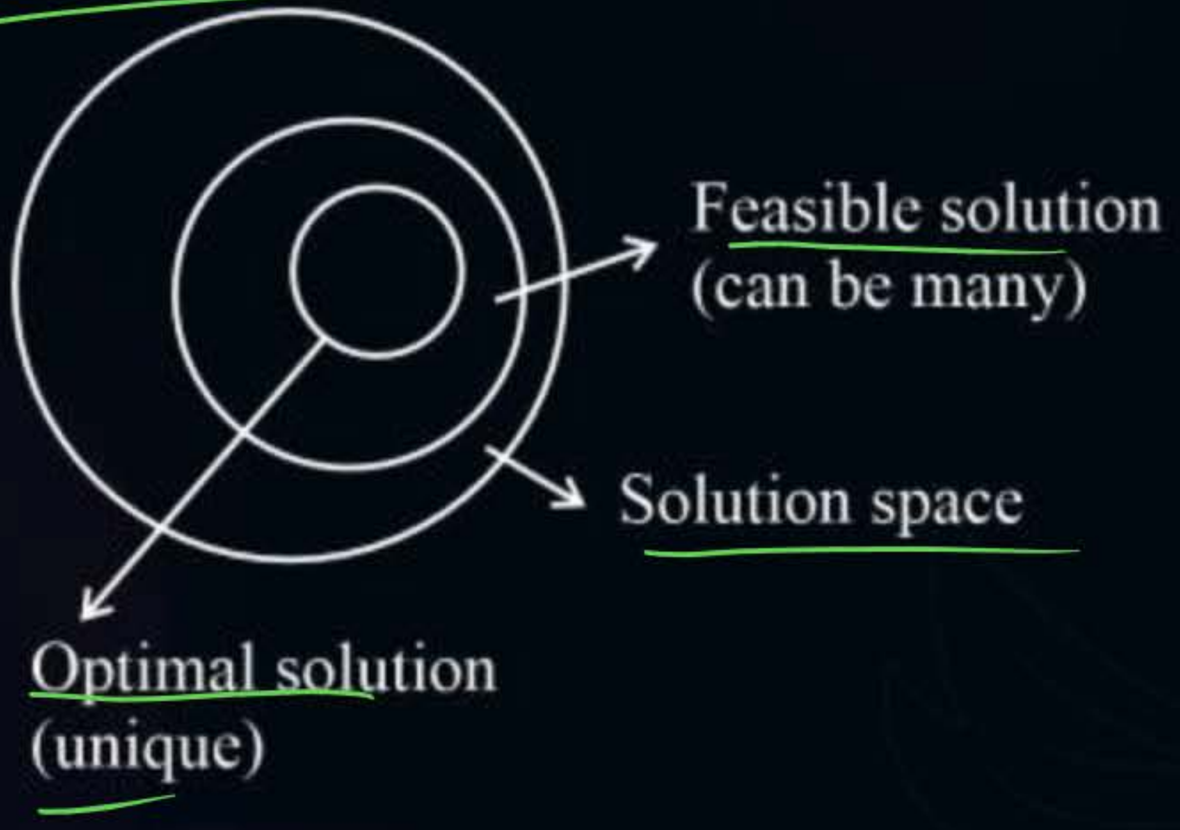
A → B → C : 5 + 7 = 12

A → D → C : 8 + 4 = 12

12

[NAT]

#Q. What is the objective function of n-Queens problem?



- It does not have any objective function. Hence there is no optimal solution.
- It has multiple feasible solutions.

Topic : Greedy Algorithm

Problem (P)

Decision Problem

- result is always either True/False.
- Feasible Solution(s)

e.g.: Searching, Sorting, n-Queens etc.

Optimization Problem

- Requires to determine a min/max value (optimal sol) for a given criteria

[redacted]

- Objective function

e.g.: Knapsack
Shortest Path Problem

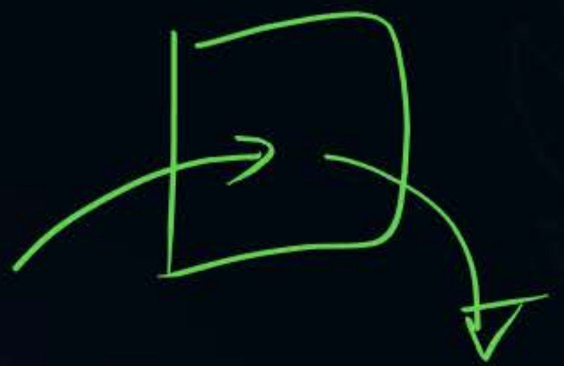
JSD

Problem 2:

Job Sequencing with Deadline (JSD):

(default)

- Given a single CPU, Non-Preemptive Scheduling.
- Given n-jobs / Tasks/Processes



Assignments

$A1 \rightarrow m_1 \rightarrow D_1$

$A_2 \rightarrow m_2 \rightarrow D_2$

$A3 \rightarrow m_3 \rightarrow D_3$

Topic : Greedy Algorithm

Description:

Job i (i = 1 to n)

- Arrival time (AT) ↓ 0
- Burst time (BT) ↓ 1
- Deadline (D_i)
- Profit (P_i)

Topic : Greedy Algorithm

Problem Statement:

- Select a subset of 'n' given jobs, such that all the jobs in the subset are completed within their corresponding deadlines and generate the maximum profit.

Note: Only when the job gets completed with the given deadline, then only you get the associated profit (else not)

Topic : Greedy Algorithm

Example:

$n = 4, J = \phi, |J| = 0$, Profit = 0

Jobs	**Profit**	**Deadline**
J1	200	2
J2	30	1
J3	50	2
J4	80	1

$J = \phi$

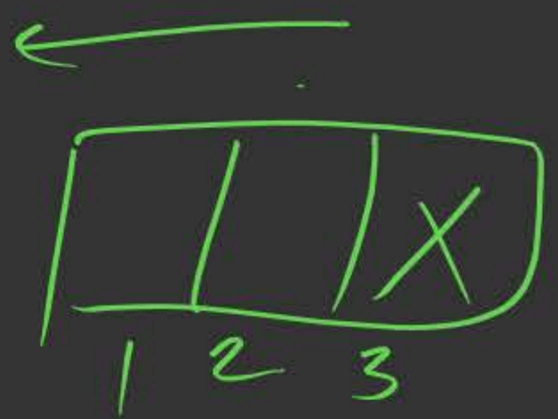
$|J| = 1$

$\underset{1}{\boxed{J1}}$, Profit = $\underline{\underline{200}}$

$|J| = 2$ $\quad \begin{array}{|c|c|} \hline J_4 & J_1 \\ \hline \end{array}$ $\Rightarrow$ Profit = $80 + 200$

$= \underline{\underline{280}}$ ✓

1	2



Topic : Greedy Algorithm

- This approach is Brute Force (Enumeration)

  n Jobs → Size of solution space → $O(2^n)$

  Hence, not a good idea to solve like this.

- We need a Greedy Algorithm based approach.

$n \longrightarrow 2^n$

Topic : Greedy Algorithm

Example:

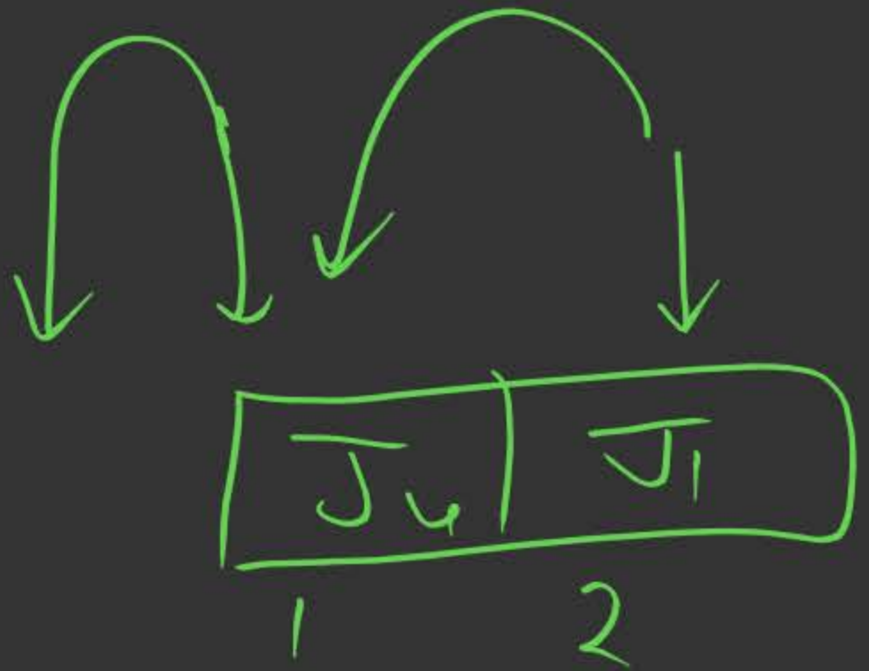
[redacted]

Jobs	Profit	Deadline
J1	200	2
J2	30	1
J3	50	2
J4	80	1

Priority
1 ✓
4 ✗
3 ✗
2 ✓

d_i

$max(D_i) = 2$



maxProfit = 200 + 80

= 280

2 mins Summary

Topic

Topic

Topic

Topic

Topic

Knapsack

JSD

THANK - YOU