

# CS & IT ENGINEERING

## Algorithms

Miscellaneous Topics

Lecture No. - 02

By- Dr. Khaleel Khan  
sir





# Recap of Previous Lecture



**Topic**

**Introduction to Greedy Method**

**Control Abstraction**

**Knapsack Problem**



# Topics to be Covered



**Topic**

**Knapsack Problem**

**Job Sequencing with Deadlines**

**Optimal Merge Patterns**







## Topic : Greedy Method



$$n=3; M=20; (P_1, P_2, P_3) = \langle 25; 24; 15 \rangle$$
$$(w_1, w_2, w_3) = \langle 18; 15; 10 \rangle$$

(i) Greedy about Profit: 28.3

(ii) Greedy about weight: 31

(iii) Greedy about Profit Per unit weight:

$$\begin{array}{lcl} w_i & \longrightarrow & P_i \\ 1 & \longrightarrow & ? \\ & & P_i/w_i \end{array}$$

$$\frac{P_1}{w_1} = \frac{25}{18} \approx 1.38 \quad \therefore x_1 = 0$$

$$\frac{P_2}{w_2} = \frac{24}{15} = 1.6 \checkmark \quad \therefore x_2 = 1$$

$$\frac{P_3}{w_3} = \frac{15}{10} = 1.5 \quad \therefore x_3 = 5/10 = 1/2$$

$$\sum w_i x_i = 20;$$

$$\sum P_i x_i = 0 + 24 + \frac{1}{2} \times 15 = 31.5$$





## Topic : Greedy Method



- a) Find an optimal solution to the knapsack instance  $n = 7$ ,  $M = 15$ ,  $(p_1, p_2, \dots, p_7) = (10, 5, 15, 7, 6, 18, 3)$  and  $(w_1, w_2, \dots, w_7) = (2, 3, 5, 7, 1, 4, 1)$ ?

$$\checkmark \frac{p_1}{w_1} = \frac{10}{2} = 5$$

$$\frac{p_2}{w_2} = \frac{5}{3} = 1.66$$

$$\checkmark \frac{p_3}{w_3} = \frac{15}{5} = 3$$

$$\frac{p_4}{w_4} = \frac{7}{7} = 1$$

$$\checkmark \frac{p_5}{w_5} = \frac{6}{1} = 6$$

$$\checkmark \frac{p_6}{w_6} = \frac{18}{4} = 4.5$$

$$\checkmark \frac{p_7}{w_7} = \frac{3}{1} = 3 \checkmark$$

$$M = 15$$

$$x_5 = 1$$

$$x_1 = 1$$

$$x_6 = 1$$

$$x_3 = 1$$

$$x_7 = 1$$

$$x_2 = \frac{2}{3}$$

$$\left\langle 1 + 2 + 4 + 5 + 1 + \frac{2}{3} \times 3 \right\rangle$$

$$\text{Profit: } 55.33$$





## Topic : Greedy Method



Procedure GREEDY\_KNAPSACK(P, W, M, X, n)

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

// objects ordered so that  $P(i)/W(i) \geq (P + i)/w(l + 1)$ . M is the //

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

real P(1 : n), W(l : n), X(1: n), M, cu;

integer l, n;

X  $\leftarrow$  0 //initialize solution to zero//

cu  $\leftarrow$  M //cu = remaining knapsack capacity//

for i  $\leftarrow$  1 to n do

if W(i) > cu then exit endif

X(i)  $\leftarrow$  1

cu  $\leftarrow$  cu - W(i)

repeat

if i  $\leq$  n then X(i)  $\leftarrow$   $(cu/W(i))$  endif

end GREEDY—KNAPSACK

Time:  $O(n)$

If Sorting is Considered

then :  $O(n \log n + n)$

$O(n \log n)$





# Topic : III. Greedy Method

$$M = 11$$

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

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

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

Greedy Strategy

$$\frac{P_1}{W_1} = \frac{60}{10} = 6$$

$$\frac{P_2}{W_2} = \frac{28}{7} = 4$$

$$\frac{P_3}{W_3} = \frac{20}{4} = 5$$

$$\frac{P_4}{W_4} = \frac{24}{2} = 12$$

$\langle 2+4 \rangle$

$$\begin{cases} x_4 = 1 \\ x_3 = 1 \end{cases}$$

Profit: 44

$$V_{Greedy} = 44$$

✓ greedy :

$$x_4 = 1$$

$$x_1 = 9/10$$

$$\langle 2 + \frac{9}{10} \times 10 \rangle = 11 \checkmark$$

$$\langle 24 + \frac{9}{10} \times 60 \rangle = \textcircled{78} \checkmark$$





## Topic : III. Greedy Method

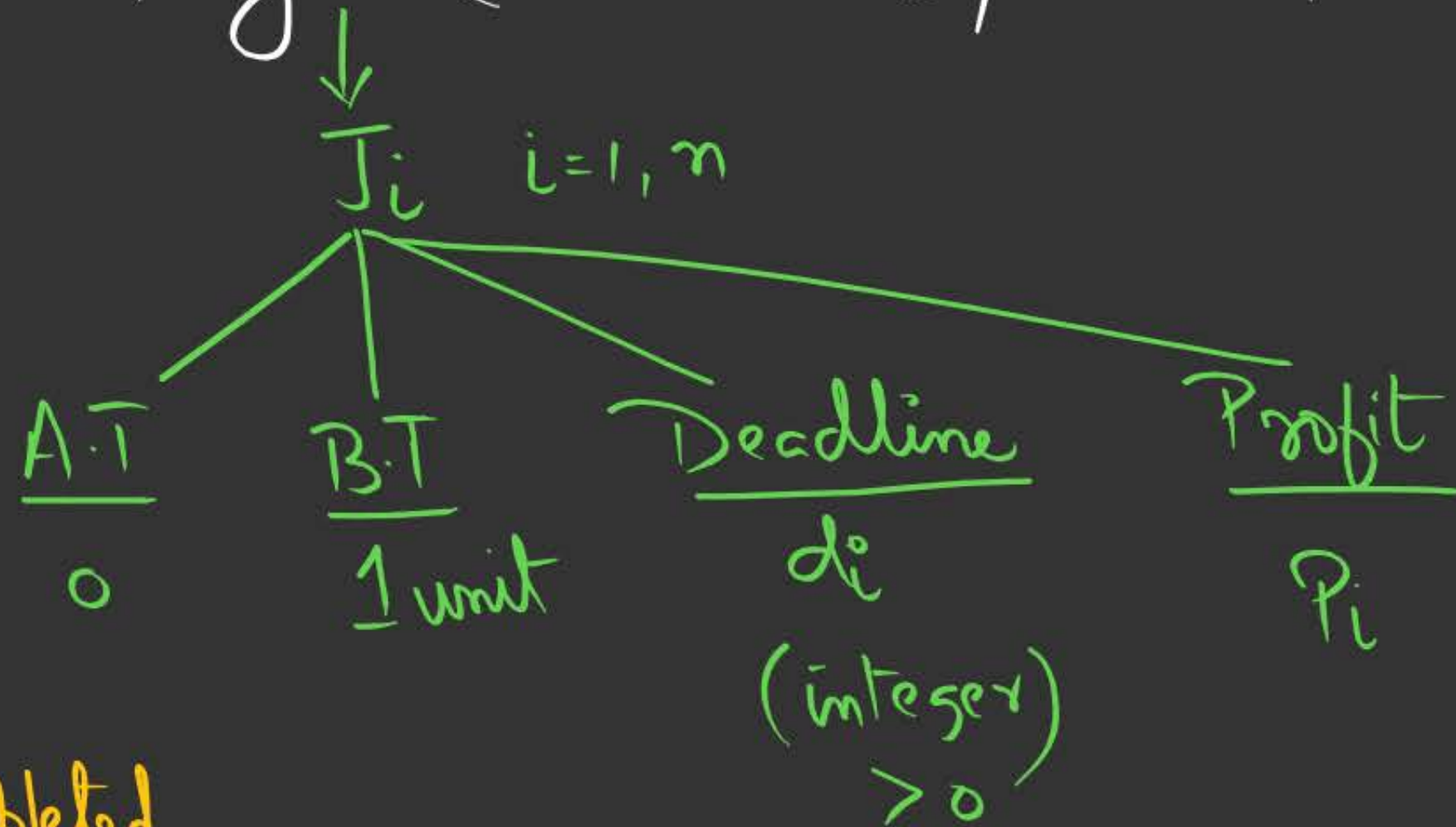
Q. The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by  $V_{\text{opt}}$ . A greedy algorithm sorts the items by their 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  $V_{\text{greedy}}$ . The value of  $V_{\text{opt}} - V_{\text{greedy}}$  is  $60 - 44 = 16$  ✓



## 2) Job-Sequencing with deadlines (JSD) [CPU-Scheduling Problem]

→ Given a Single CPU, using Non-Preemptive Scheduling;

→ Given a Set of  $n$ -jobs (Processes)/Tasks/Programs



A.T = Arrived Time

B.T = Burst Time

→ If the job is completed within the deadline, then you get its profit



Prob Defn: Select a Subset of 'n' given jobs, Such that the jobs in the Subset are Completable within their deadlines and generate maximum Profit;

Memory  $(J_1 J_2 J_3 \dots J_n)$

OS Scheduler

→ Select a Subset

→ Feasibility criteria

Implicit Constn:

< Completing jobs in the Subset within their deadlines >

Objective fn:

Max Profit

Q) Size of Soln Space:

$$\text{No. of Subsets} = \{2^n\}$$

CPU

Subset paradigm



→ Assignment Problem:

$c_1$	—	$d_1$	—	10	
$c_2$	—	$d_2$	—	15	✓✓
$c_3$	—	$d_3$	—	5	
$c_4$	—	$d_4$	—	30	✗
$c_5$	—	$d_5$	—	15	
$c_6$	—	$d_6$	—	20	



1)  $n=4; \langle J_1 \dots J_4 \rangle$   
 $\langle d_1 \dots d_4 \rangle = \langle 2, 1, 2, 1 \rangle$   
 $\langle p_1 \dots p_4 \rangle = \langle 100; 15; 25; 40 \rangle$

A.T=0  
B.T=1

$J \leftarrow \emptyset$

1)  $|J| = \text{zero (feasible)}$

2)  $|J| = 1$


$J \leftarrow \{J_1\}$  

$J \leftarrow \{J_2\}$


All subsets with only 1 job are feasible

3)  $|J| = 2$

$J \leftarrow \{J_1, J_2\} : \text{feasible}$

cpu  115

$J \leftarrow \{J_1, J_3\} : \text{feasible}$

cpu   
(Any order)

$J \leftarrow \{J_2, J_4\} \times$

cpu  Not feasible

$J \leftarrow \{J_3, J_4\}$

order:  $\langle J_4, J_3 \rangle$

$|J| = 3$

$J \leftarrow \{J_1, J_2, J_3\}$

  
0 1 2 3

$L = 3$

Max Jobs in a Feasible Subset =  $\max(d_i)$   
 $i=1, n$

$B.T(J_i) = 1$



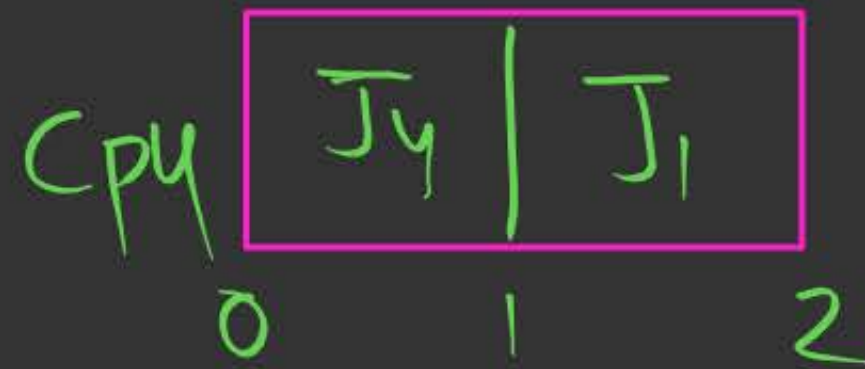
$$n=4; \langle J_1 \dots J_4 \rangle$$

$$\langle d_1 \dots d_4 \rangle = \langle 2, 1, 2, 1 \rangle$$

$$\langle P_1 \dots P_4 \rangle = \langle \underline{100}; \underline{15}; \underline{25}; \underline{40} \rangle$$

$\times \quad \times$

$J \leftarrow \{$



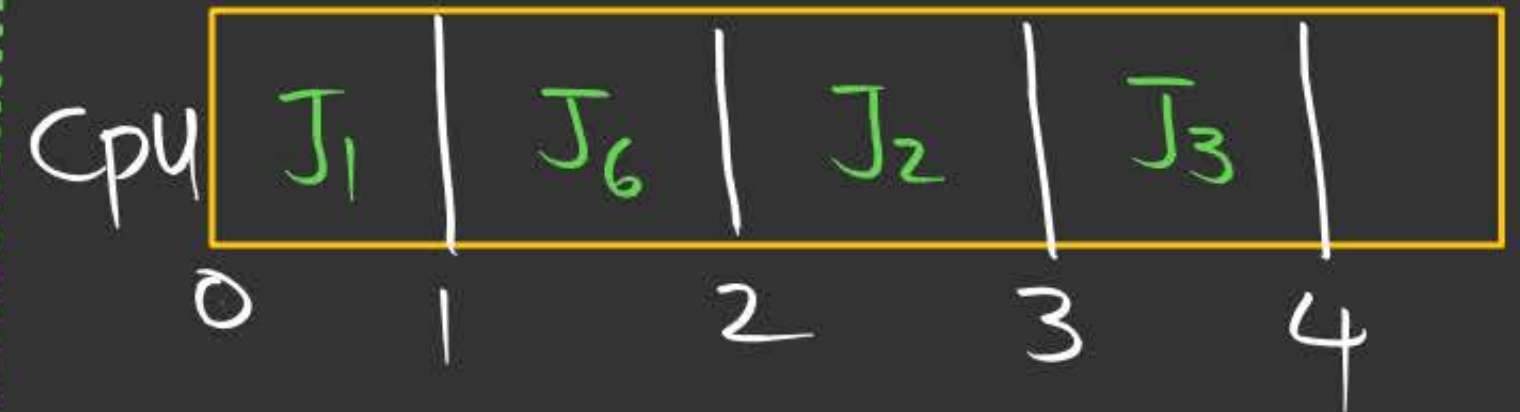
Final Profit: 140

$$2) \quad n=6; \langle J_1 \dots J_6 \rangle$$

$$\langle d_1 \dots d_6 \rangle = \langle 2, 3, 4, 2, 4, 3 \rangle$$

$$\langle P_1 \dots P_6 \rangle = \langle \underline{14}; \underline{18}; \underline{12}; \underline{10}; \underline{8}; \underline{15} \rangle$$

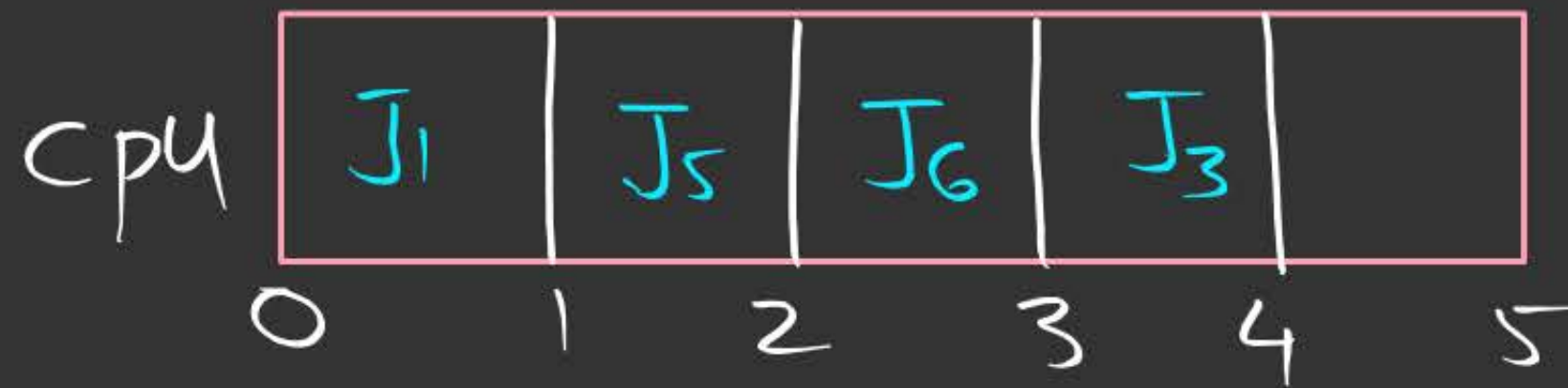
$\cdot \quad \cdot \quad \cdot \quad \times \quad \times \quad \cdot$



Profit: 59



3)  $n=6; \langle J_1 \dots J_6 \rangle$   
 $\langle d_1 \dots d_6 \rangle : \langle \bar{2}, 1, 5, \bar{1}, \underline{2}, 3 \rangle$   
 $\langle p_1 \dots p_6 \rangle : \langle \underline{28}, \underline{12}, 5, \underline{18}, \underline{30}, \underline{20} \rangle$   
 $\times$   $\times$



Final Profit : 83



4)  $m=7$ ;  $\langle J_1 - J_2 \rangle$

$$\langle d_1 \dots d_7 \rangle = \langle 4; 4; 4; 4; \underline{3}; 4; 3 \rangle$$

$$\langle p_1 \dots p_7 \rangle = \langle \underline{70}; \underline{85}, 12; 18; \underline{\underline{50}}; \underline{\underline{60}}; 10 \rangle$$



Profit: 265





## Topic : Greedy Method



1. Algorithm Greedy Job(d, J, n)
2. // J is a set of jobs that can be completed by their deadlines. *Assuming Jobs are in decreasing order of Profits*
3. {
4.   J := {1};
5.   for i := 2 to n do :  $O(n)$
6.   {
7.     if (all jobs in J U {i} can be completed
8.       by their deadlines) then J := J U {i};
9.   }
10. }

$O(n)$  } Time Complexity :  $O(n^2)$

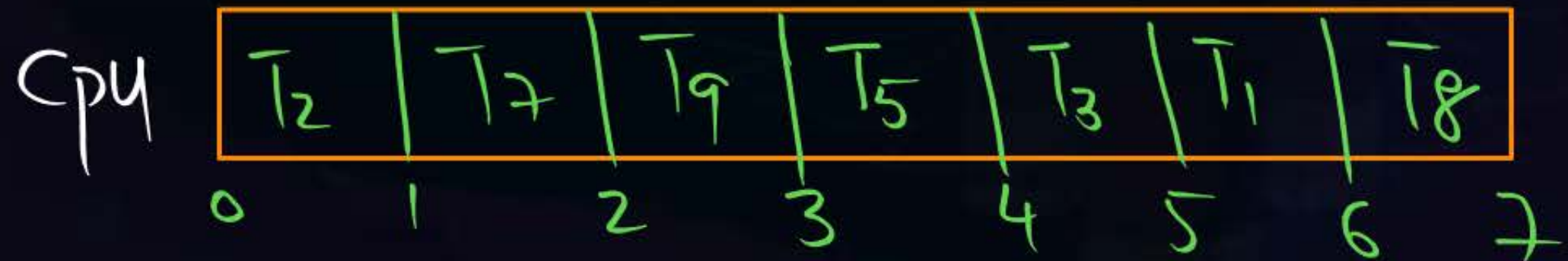




## Topic : III. Greedy Method

- Q. We are given 9 tasks  $T_1, T_2, \dots, T_9$ . The execution of each task requires one unit of time. We can execute one task at a time. Each task  $T_i$  has a profit  $P_i$  and a deadline  $d_i$ , Profit  $P_t$  is earned if the task is completed before the end of the Deadline.

Task	<u>T1</u>	<u>T2</u>	<u>T3</u>	<del>T4</del>	<u>T5</u>	<del>T6</del>	<u>T7</u>	<u>T8</u>	<u>T9</u>
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3







## Topic : III. Greedy Method

- a. Are all tasks completed in the schedule that gives maximum profit?
- (a) All tasks are completed
  - (b) T1 and T6 are left out
  - (c) T1 and T8 are left out
  - ✓ (d) T4 and T6 are left out
- b. What is the maximum profit earned?
- ✓ (a) 147
  - (b) 165
  - (c) 167
  - (d) 175



### 3) Optimal Merge Patterns: <Ordering Paradigm>

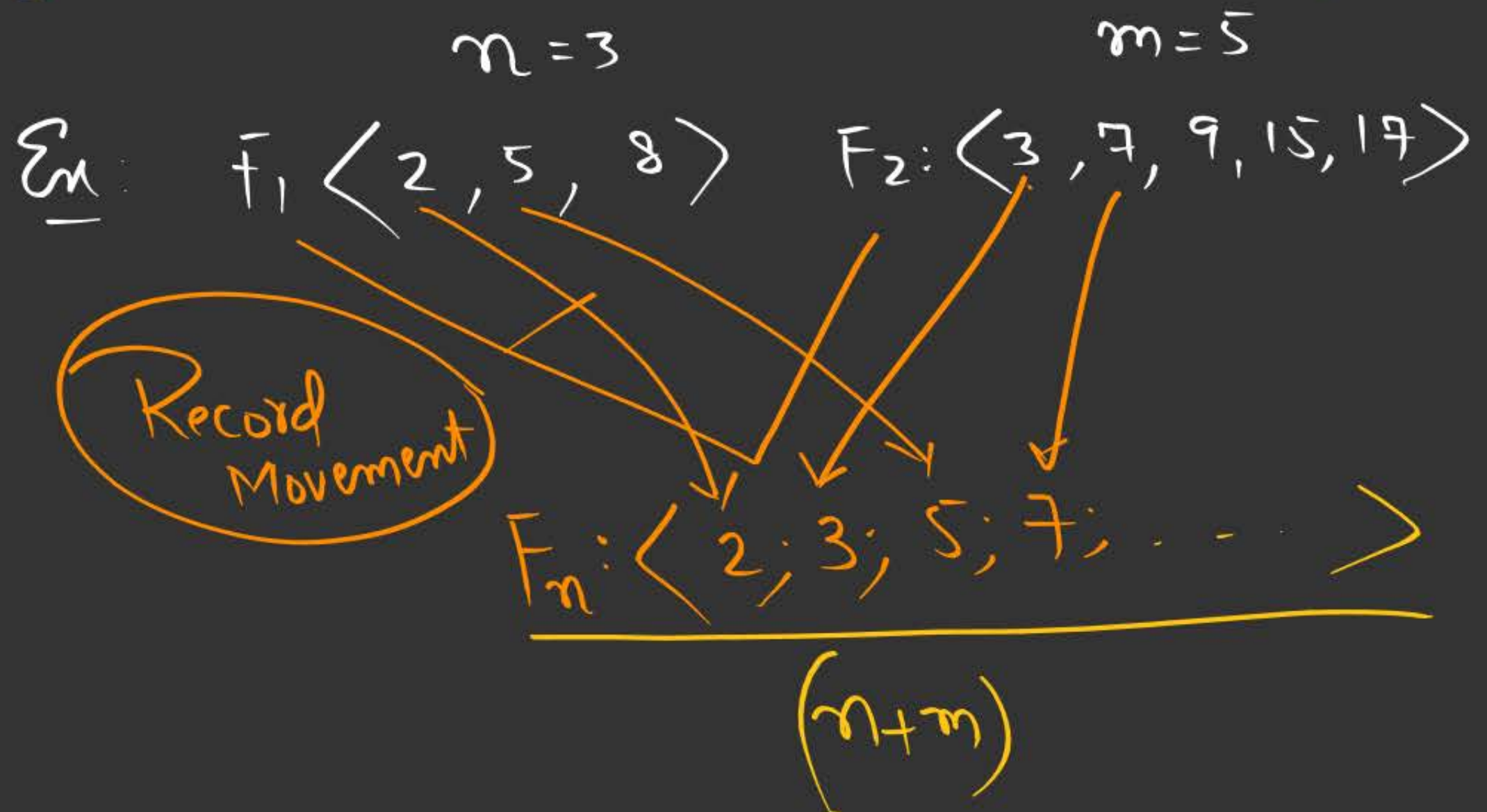
→ Given a Set of  $n$ -Files  $\langle F_1, F_2, F_3, \dots, F_n \rangle$

Each File Contains a Set of records in Sorted order;

It is required to Merge the given  $n$ -Files,  
to get a Single File in Sorted order, using 2-way  
Merging;

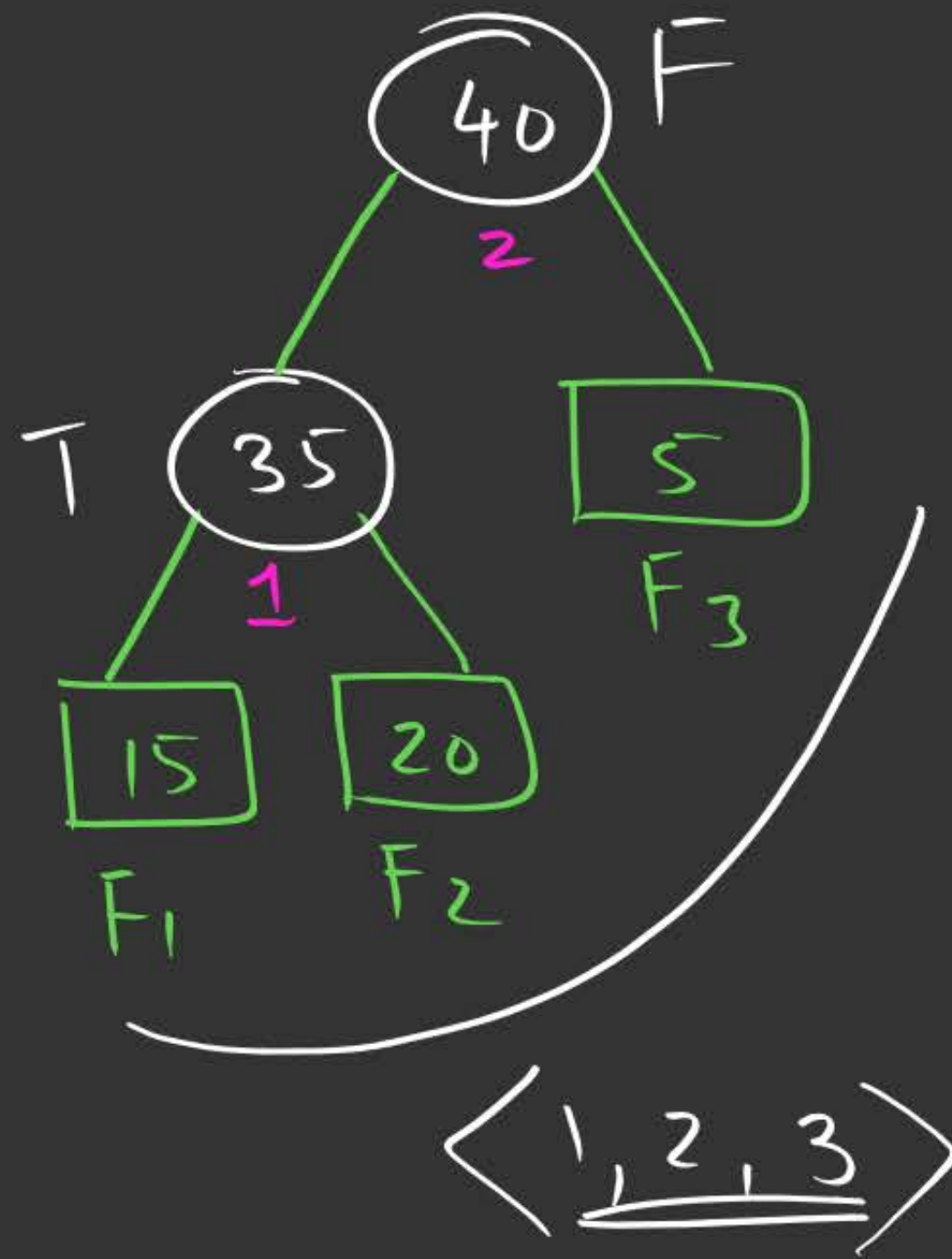
Metric:

No of record  
movements

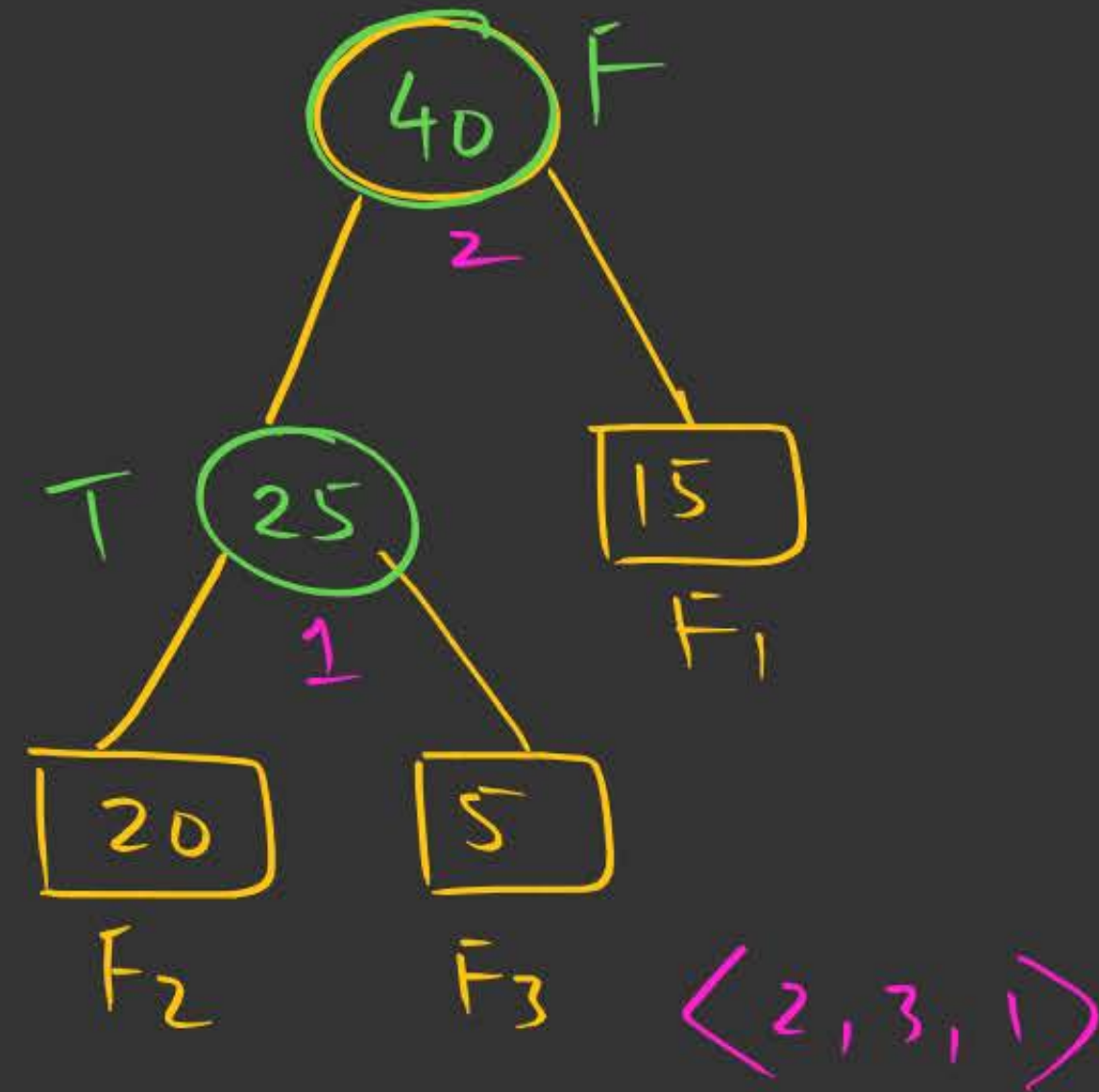




$$F_1 = \overline{15}; \quad F_2 = \overline{20}; \quad F_3 = \overline{5}$$



$$\text{Total rec. mov's} : 35 + 40 = \underline{\underline{75}}$$



$$\text{Total Record mov's} : 65 = \underline{\underline{65}}$$



$\Rightarrow$  we want that pattern that generates Minimum  
No. of record Movements (Optimal)

Q) Soln Space :  $n!$        $\langle F_1 \dots F_n \rangle$



**THANK - YOU**