CS & IT ENGINEERING

Algorithms

Greedy Method

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 :

$w_i \longrightarrow P_i$

$1 \longrightarrow ?$

P_i / w_i

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

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

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

$\sum w_i x_i = 20;$

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

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

✓ $\frac{P_1}{W_1} = \frac{10}{2} = 5$

$\frac{P_2}{W_2} = \frac{5}{3} = 1.66$

✓ $\frac{P_3}{W_3} = \frac{15}{5} = 3$

$\frac{P_4}{W_4} = \frac{7}{7} = 1$

✓ $\frac{P_5}{W_5} = \frac{6}{1} = 6$

✓ $\frac{P_6}{W_6} = \frac{18}{4} = 4.5$

✓ $\frac{P_7}{W_7} = \frac{3}{1} = 3$ ✓

$M = 15$

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

$x_5 = 1$

$x_1 = 1$

$x_6 = 1$

$x_3 = 1$

$x_7 = 1$

$x_2 = 2/3$

$\boxed{\text{Profit: } 55.33}$

```
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) ≥(P + i)/w(I + 1). M is the//
// knapsack size and X(I : n) is the solution vector//
real 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
        cu ← cu – W(i)
repeat
        if i ≤ n then X(i) ← cu/W(i) endif
end GREEDY—KNAPSACK
```

fraction

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

$V_{opt} = 60$

Greedy Strategy

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

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

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

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

$\langle 2+4 \rangle$

$\left\{ \begin{matrix} x_4 = 1 \\ x_3 = 1 \end{matrix} \right\}$

Profit: 44

$V_{Greedy} = 44$

V_{greedy}:

$x_4 = 1$

$x_1 = 9/10$

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

$\langle 24 + \frac{9}{\cancel{10}} \times \cancel{60}6 \rangle = 78$ ✓

Slide 7

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, $\boxed{\text{no item may be split.}}$ 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 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

↓

J_i $\quad i = 1, n$

A.T	B.T	Deadline	Profit
0	1 unit	d_i (integer) > 0	P_i

A.T = Arrival 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 ($J_1\ J_2\ J_3 \ldots J_n$)

Select a Subset

OS Schdler

CPU

Subset paradigm

Feasibility criteria

Implicit Constn :

<completing jobs in the subset within their deadlines>

Objective fn :

Max. Profit

Q) Size of Soln Space :

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

$\rightarrow$ **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) $n = 4$; $\langle J_1 .. J_4 \rangle$

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

$\langle P_1 .. P_4 \rangle = \langle 100; 15; 25; 40 \rangle$

A.T = 0

B.T = 1

$J \leftarrow \phi$

1) $|J|$ = zero (feasible)

2) $|J| = 1$

$J \leftarrow \{J_1\}$

cpu

J_1	

0 1

$J \leftarrow \{J_2\}$

:

All subsets with only 1 job are feasible

3) $|J| = 2$

$J \leftarrow \{J_1, J_2\}$: Feasible

115

cpu

J_2	J_1	

0 1 2

$J \leftarrow \{J_1, J_3\}$: Feasible

cpu

J_1	J_3

0 1 2

(Any order)

$J \leftarrow \{\overset{1}{J_2}, \overset{1}{J_4}\}$ ×

NOT Feasible

cpu

	×

0 1 2

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

order: $\langle J_4; J_3 \rangle$

$|J| = 3$

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

0 1 2 3

$\ell = 3$

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

$B.T(J_i) = 1$

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

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

$\langle P_1 \ldots P_4 \rangle = \langle 100; 15; 25; 40 \rangle$

x x

$J \leftarrow \{$

CPU	J_4	J_1

0 1 2

Final Profit: 140

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

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

$\langle P_1 \ldots P_6 \rangle = \langle 14; 18; 12; 10; 8; 15 \rangle$

x x

CPU	J_1	J_6	J_2	J_3	

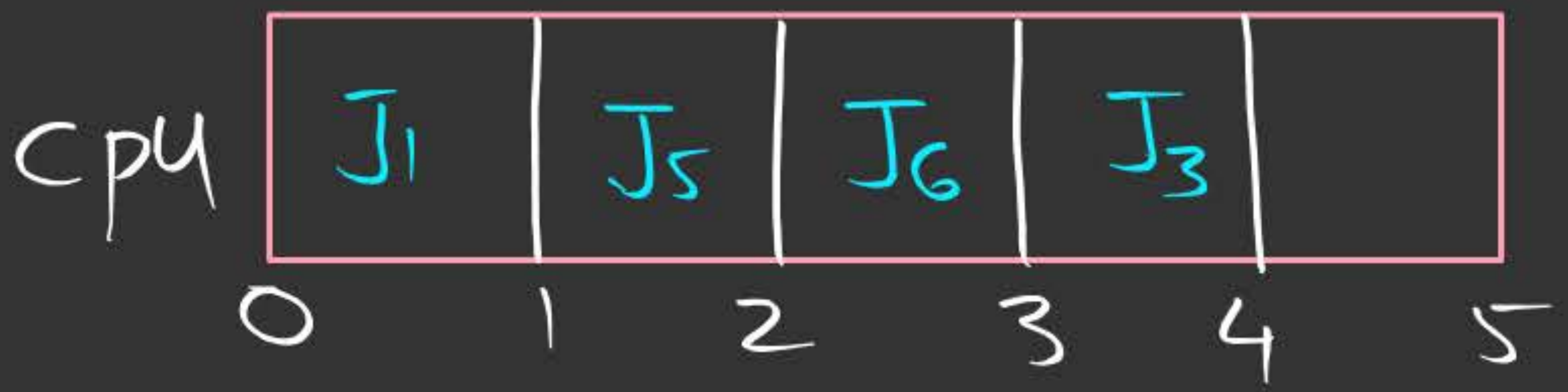
0 1 2 3 4

Profit: 59

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

$\langle d_1 \dots d_6 \rangle : \langle 2, 1, 5, 1, 2, 3 \rangle$

$\langle P_1 \dots P_6 \rangle : \langle 28; 12; 5; 18; 30; 20 \rangle$



Final Profit : 83

4) $n = 7$; $\langle J_1 - J_7 \rangle$

$\langle d_1 \ldots d_7 \rangle = \langle 4; 4; 4; 4; 3; 4; 3 \rangle$

$\langle P_1 \ldots P_7 \rangle = \langle 70; 85, 12; 18; 50; 60; 10 \rangle$

CPU	J_5	J_6	J_1	J_2	
	0 – 1	1 – 2	2 – 3	3 – 4	

Profit: 265

Topic : Greedy Method

```
Algorithm Greedy Job(d, J, n)
// J is a set of jobs that can be completed by t heir
    deadlines.
{
    J :={1};
    for i := 2 to n do
    {
        if (all jobs in J U {i} can be completed
                by their deadlines) then J: = J U {i};
    }
}
```

Assuming Jobs are in decreasing order of Profits

for i := 2 to n do : $O(n)$

$O(n)$

Time Complexity : $O(n^2)$

Topic : III. Greedy Method

Q. We are given 9 tasks T1, T2...., T9. The execution of each task requires one unit of time. We can execute one task at a time. Each task T1 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	T1	T2	T3	T4	T5	T6	T7	T8	T9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

cpu

T_2	T_7	T_9	T_5	T_3	T_1	T_8

0 1 2 3 4 5 6 7

Slide 10

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;

$n = 3$ $\quad\quad$ $m = 5$

Ex: $F_1 \langle 2, 5, 8 \rangle$ $\quad$ $F_2 : \langle 3, 7, 9, 15, 17 \rangle$

Metric:

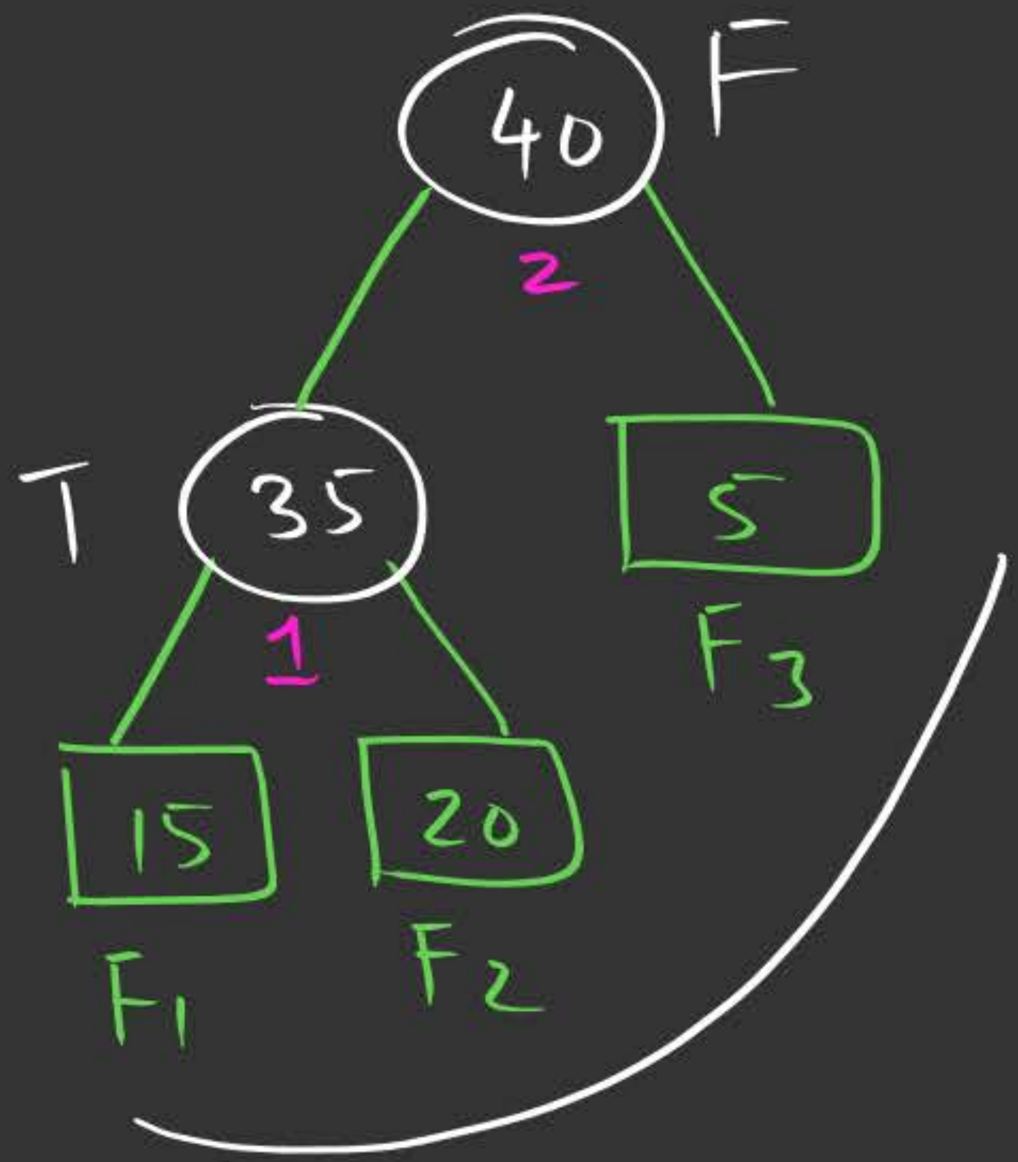
No of record movements

Record Movement

$F_n : \langle 2; 3; 5; 7; \dots \rangle$

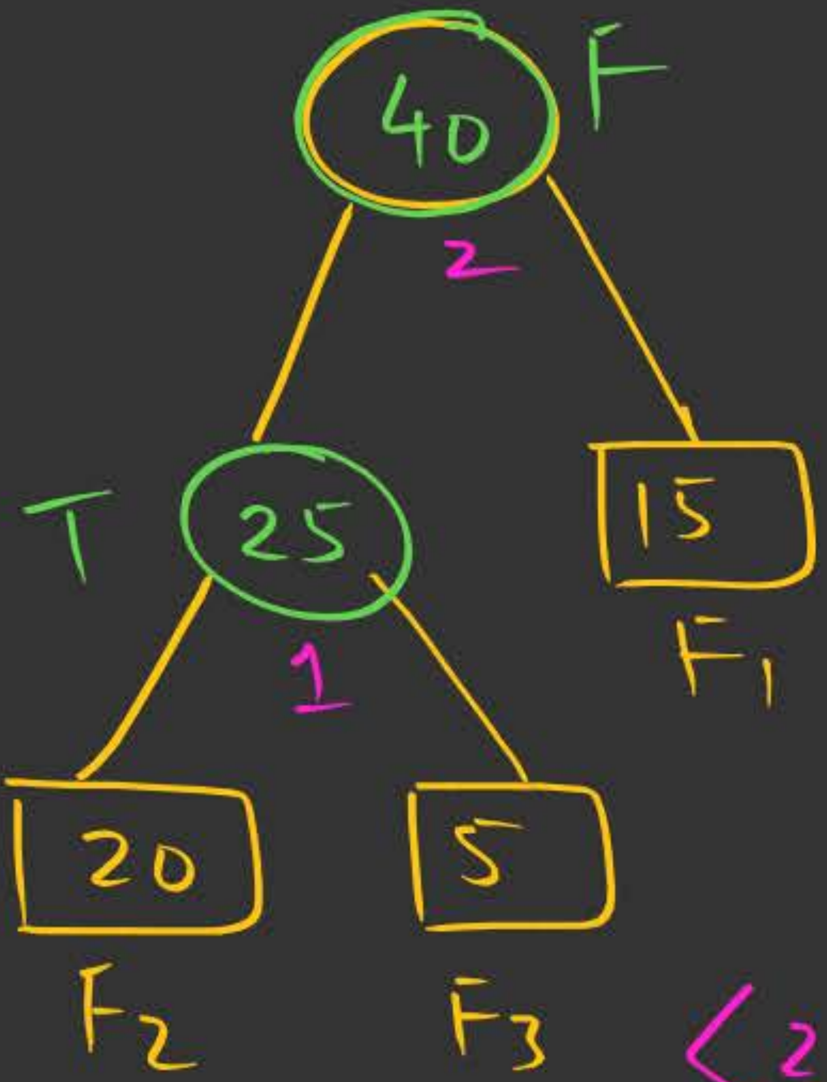
$(n+m)$

$F_1 = 15; \quad F_2 = 20; \quad F_3 = 5$



$\langle 1, 2, 3 \rangle$

Total rec. mov's : $35 + 40 = 75$



$\langle 2, 3, 1 \rangle$

Total Record mov's : 65

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

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

THANK - YOU