Day: Date: / /

$8^{\log n} T\left(\frac{n}{2^{\log n}}\right) + \left(2^{\log n} - 1\right) n^2$

$8^{\log n} . T\left(\frac{n}{2^{\log n}}\right) + \left(2^{\log n} - 1\right) n^2$

$n^{\log 8} : T\left(\frac{n}{n^{\log 2}}\right) + \left(n^{\log_2 2} - 1\right) n^2$

$n^3 T(\cancel{n}/\cancel{n}) + (n-1) n^2$

$n^3 T(1) + n^3 - n^2$

$= n^3 + n^3 - n^2 = 2n^3 - n^2 = O(n^3)$

Strassan's Multiplication:

1. Reduce No of Multiplication i.e = 7
2. $P = (A_{11} + A_{22})(B_{11} + B_{22})$ $S = A_{22}(B_{21} - B_{11})$

$Q = (A_{21} + A_{22}) \cdot B_{11}$ $T = (A_{11} + A_{12}) B_{22}$

$R = A_{11}(B_{12} - B_{22})$ $U = (A_{22} - A_{11}) \cdot (B_{11} + B_{12})$

$V = (A_{12} - A_{22})(B_{21} + B_{22})$

$C_{11} = P + S - T + V$ $C_{21} = Q + S$

$C_{12} = R + T$ $C_{22} = P + R - Q + U$

$T(n) = 7T(n/2) + n^2 \longrightarrow O(n^{\log 7}) = O(n^{2.81})$

Greedy Paradigm Approach:

1. a design for solving a problem.
2. used for optimization problem → (i) Maximisation (ii) Minimization
3. It has objective and constraints.
4. Solution which meets the constraints to complete objective is Feasible solution.
5. optimal solution will always be 1.

Prince

Day: Date: / /

cost effective 10 hrs

Sialkot ⟶ Peshawar

walk | bike | car | train | flight

feasible solution

optimal solution

→ Minimization problem as we are minimizing the cost and time.

1. Dynamic Programming
2. Branch and Bound

} can also be used for optimal solution.

→ 0/1 Knapsack Problem
→ Fractional Knapsack Problem

→ **Knapsack Problem:**

			x_1	x_2	x_3	x_4	x_5	x_6	x_7
			1	2/3	1	0	1	1	1
objects	:	O	1	2	3	4	5	6	7
Profit	:	P	10	5	15	7	6	18	3
weight	:	W	2	3	5	7	1	4	1

2

15 - 4 = 11

11 - 5 = 6

constraints: Do not exceed the capacity of Bag = 15 kg

objective: Gives maximum Profit.

P/W	5	1.67	3	1	6	4.5	3

Bag capacity = m = 15

$m - 1 = 14$, $2 - m - 14 - 2 = 12$, $12 - 4 = 8$, $8 - 5 = 3$, $3 - 1 = 2$

$2 - 2 = 0$

$x_i w_i = (1\times2) + (\frac{2}{3}\times3) + (1\times5) + (0\times7) + (1\times1) + (1\times4) + (1\times1)$

$x_i w_i = 2 + 2 + 5 + 0 + 1 + 4 + 1$

$x_i w_i = 15$

constraint → $\Sigma x_i w_i \le m$ ∵ $m = 15\,kg$

Profit $= x_i P_i = (1\times10) + (\frac{2}{3}\times5) + (1\times15) + (0\times7) + (1\times6) + (1\times18) + (1\times3)$

$x_i P_i = 10 + 10/3 + 15 + 0 + 6 + 18 + 3$

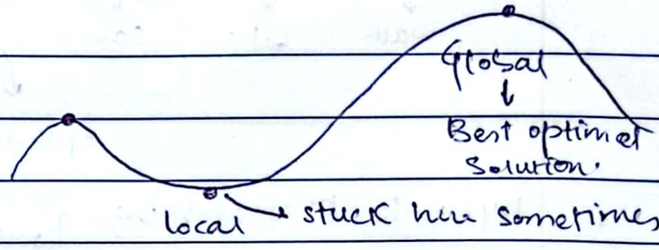
$\frac{10}{15}$ $x_i P_i = 55.3$

Weight of Items / 15 - 10

Prince

Day: Date: / /

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Based on profit:	1	0	1	4/7	0	1	0



$\because m = 15$

$15 - 4 = 11$

$11 - 5 = 6$

$6 - 2 = 4$

$4 - 4 = 0$

$x_i P_i = (1\times10) + (0) + (1\times15) + (\frac{4}{7}\times 7) + (0) + (1\times18) + (0)$

$= 10 + 15 + 4 + 18$

$\underline{= 47}$. Ans

$$\begin{array}{r} 29 \\ 18 \\ \hline 47 \end{array}$$

→ Previous approach on the basis of weight is more optimal for maximizing profit

x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1	4/5	0	1	1	1

$\therefore m = 15$

$15 - 1 = 14$

$14 - 1 = 13$

$13 - 2 = 11$

$11 - 3 = 8$

$8 - 4 = 4$

$4 - 4 = 0$

$x_i P_i = (1\times10) + (1\times5) + (\frac{4}{5}\times \overset{3}{\cancel{15}}) + (0) + (1\times6) + (1\times18) + (1\times3)$

$= 10 + 5 + 12 + 6 + 18 + 3$

$\boxed{= 54}$

$$\begin{array}{r} 36 \\ 18 \\ \hline 54 \end{array}$$

Hello guys and gays

Prince

Day: Date: / /

New → Job sequencing with deadlines :

Jobs	J	J_1	J_2	J_3	J_4	J_5
Profit	P	20	15	10	5	1
Deadline	D	2	2	1	3	3

$\underline{J_2}$ 1 $\underline{J_1}$ 2 $\underline{J_4}$ 3 ~~J_5~~

20 + 15 + 5 = 40

Job	slots	solution	Profit
J_1	[1,2]	$\{J_1\}$	20
J_2	[0,1][1,2]	$\{J_1, J_2\}$	20+15
J_3	[0,1][1,2]	$\{J_1, J_2\}$	20+15
J_4	[0,1][1,2][2,3]	$\{J_1, J_2, J_3\}$	20+15+5
J_5	[0,1][1,2][2,3]	$\{J_1, J_2, J_3\}$	20+15+5

1) We can't Backtrack in Greedy Algorithm.

2) We may get stuck in Local minima in Greedy.

Activity Selection:

→ Given activities (lectures) and one room, Now schedule as many activities as possible, such that they do not conflict each other.

Hehehuhuhaha

Hello guys

Kia?

Prince

Bad boy Ignore Us ne baat ki thi kal

	i	1	2	3	4	5	6	7	8	9
	Si	2	2	4	1	5	8	9	11	13
	Fi	3	5	7	8	9	10	11	14	16
duration.	di	1	3	3	7	4	2	2	3	3
By difference	App 1	✓	✗	✓	✗	✗	✓	✗	✓	✗
By low Ending Time	App 2	✓	✗	✓ ✗	✗	✗	✓	✗	✓	✗
	Max st	✗			✗	✓	✗	✓	✗	✓

Total subsets $= 2^9$

Solution:

$\{a_1, a_3, a_6, a_8\} \rightarrow$ optimal Solution.

it will be difficult to acheive optimal solution if there is more conflicts

i	=	1	2	3
si	=	4	1	5
Fi	=	6	5	10
di	=	2	4	5
		✓ ~~✗~~	✗ ~~✓~~	✗ ✓

Greedy Solution $= \{a_1\}$ App 1

optimal Solution $= \{a_2, a_3\}$ Approach 2

1. Min Duration ✗
2. Min start time ✗
3. Min End Time ✓ $\longrightarrow$ optimal Solution.
4. Max Start time

Day: Date: / /

Activity Selection Algorithm:

∵ S & F are array (S → Start, F → Finish)

∵ n = no of activities

Greedy AS(S, F , n)

//sort activities in both arrays by finish time. in ascending order.

└→ $(n \log n)$

add A[0] to A

for (i = 1 to i = n-1) → n

K = 1

~~for i = 2 to n~~

if S[i] ≥ F[K]

Add A[i] to A

K = i

Time complexity = $n + n.\log n = O(n.\log n)$