DAA ASSIGNMENT - 3

AYUSH SRIVASTAV
2193069
CSE CORE - II

1. Working of activity selection problem with gready approach.

- Sort the activities according to their finishing time.
- Select the first activity from the sorted arrary and print it.
- Do the following for remaining activities in the sorted array.
- If the start time of this activity is greater than or equal to the finish time of previously selected activity then select this activity and print it.

2.

	A_1	A_2	A_3	A_4	A_5	A_6
Start	0	3	1	5	5	8
Finish	6	4	2	9	7	9

(I) Sorting activities according to their finishing time.

	A_3	A_2	A_3	A_5	A_6	A_4
Start	1	3	0	5	8	5
finish	2	4	6	7	9	8

② Print 1st activity

∴ A3

③ Now, the starting time of activity 2 is greater than A3 so choose.

A3, A2

Similarly for A2

A3, A2, A5, A6

Total no activities = 6

And he can solve at most 4 activities in a given time.

Ans 3

$m = 15$.

Item	A	B	C	D	E	F	G
wt	2	3	5	7	1	4	1
Profit	10	5	15	7	6	18	3

$\frac{P_A}{W_A} = \frac{10}{2} = 5$, $\frac{P_B}{W_B} = \frac{5}{3} = 1.67$, $\frac{P_C}{W_C} = 3$, $\frac{P_D}{W_D} = 1$.

$\frac{P_E}{W_E} = \frac{6}{1} = 6$, $\frac{P_F}{W_F} = \frac{18}{4} = 4.5$, $\frac{P_G}{W_G} = 3$.

$\frac{P_i}{W_i} = \{5, 1.67, 3, 1, 6, 4.5, 3\}$

decreasing order of $\frac{P_i}{W_i} = \{6, 5, 4.5, 3, 3, 1.67, 1\}$

$= \{E, A, F, G, C, B, D\}$

Item choosen	Quantity	Remaining Space.	Total profit.
E	1 full item	15-1=14	6×1=6
A	1 full item	14-2=12	10×1=10
F	1 full item	12-4=8	18×1=18
G	1 full item	8-1=7	3×1=3
C	1 full item	7-5=2	15×1=15
B	2/3 of item.	2-2=0	$5 \times \frac{2}{3} = 3.33$

Total profit = 55.3

Solution set is $\{1, \frac{2}{3}, 1, 0, 1, 1, 1\}$

Ans 4.

Divide & Conqure.	Greedy Algorithm
1. Divide & Conqure is used to find the solution, it does not aim for the optimal solution	A greedy algorithm is a optimizing technique. It tries to find an optimal solution from the set of feasible.
2. DC approach is recursive in nature, so it is slower & inefficient	Greeady algorithm are iterative in nature & hence faster.
3. DC approach divides the the problem into small subproblem is solved independently & solution of the smaller problem	Greedy Algorithm does not consider the previously solved into antance again, thus in anoids the.

accombined to find the solution of the large problem	recomputations.
4. merge sort, Quick sort Binary seach, etc.	Ex. knapsact Problem, Activity selection Problem, etc.

<u>Ans 5.</u> Matrix chain multiplication is an optimization problem that to find the most efficient way to moltiply a given sequence of matrices -

. Apply matrix chain multiplication with 4 matrices

Given :-

A_1	×	A_2	×	A_3	×	A_4
$d_0 \; d_1$		$d_1 \times d_2$		$d_2 \; d_3$		$d_3 \; d_4$
3 2		2 4		4 2		2 5

formula :-

$$C[i,j] = \min_{i \le k < j} \{ C[i,k] + C[k+1,j] + d_{i-1} \times d_k \times d_j \} -$$

$$C[1,2] = \min_{1 \le k < 2} \left\{ \begin{array}{l} k=1 \\ C[1,1]_0 + C[2,2] + d_0 \times d_1 \times d_2 \end{array} \right\}$$

$$= \{ 0 + 0 + 3 \times 2 \times 4 \} = 24$$

$$C[1,2] = 24$$

$$C[2,3] = \min_{2 \le k < 3} \left\{ \begin{array}{l} C[2,2] + C[3,3]_4 + d_1 \times d_2 \times d_3 \\ 0 + 0 + 2 \times 4 \times 2 \end{array} \right\}$$

$$C[2,3] = 16$$

MIT SCHOOL OF ENGINEERING
Rajbaug, Loni-Kalbhor, Pune

$C[3,4] = \min_{3 \le k < 4} \{ C[3,3] + C[4,4] + d_2 \times d_3 \times d_4 \}$
$\{0 + 0 + 4 \times 2 \times 5\}$

$C[3,4] = 40$

$k = 1$

$C[1,3] = \min_{1 \le k < 3} \{ C[1,1] + C[2,3] + d_0 \times d_1 \times d_3 \}$
$= \{0 + 16 + 3 \times 2 \times 2$

$C[1,3] = 28$ for $k = 1$

$k = 2 \{ C[1,2] + C[3,3] + d_0 \times d_2 \times d_3 \}$
$= \{24 + 0 + 3 \times 4 \times 2\} = 48$

$C[1,3] = 48$ for $k = 2$.
$\therefore C[1,3] = 28$ for $k = 1$ as it has minimum value.

$C[2,4] = \min_{2 \le k < 4} \{ \overset{k=2}{C[2,2]} + C[3,4] + d_1 \times d_2 \times d_4 \}$
$0 + 40 + 2 \times 4 \times 5$

$C[2,4] = 80$ for $k = 2$
$k = 3 \{ C[2,3] + C[4,4] + d_1 \times d_3 \times d_4 \}$
$= 16 + 0 + 2 \times 2 \times 5$
$C[2,4] = 36$ for $k = 3$.

$\therefore C[2,4] = 36$ for $k = 3$ as it has min value.

$$C[1,4] = \min_{1 \le k < 4} \begin{cases} k=1 & C[1,1] + C[2,4] + d_0 \times d_1 \times d_4 \\ k=2 & C[1,2] + C[3,4] + d_0 \times d_2 \times d_4 \\ k=3 & C[1,3] + C[4,4] + d_0 \times d_3 \times d_4 \end{cases}$$

$$\min \left\{ \begin{array}{ccc} k=1 & k=2 & k=3 \\ 66, & 124, & 58 \end{array} \right\}$$

$C[1,4] = 58$ for $k = 3$.

k table gives us how to parethesioze.

C	1	2	3	4
1	0	24	28	58
2		0	16	36
3			0	40
4				0

K	1	2	3	4
1	x	1	1	3
2	x	x	2	3
3	x	x	x	3
4	x	x	x	x

So, the parenthization are:-

1. $((A_1 \cdot A_2), A_3) \cdot (A_4)$
2. $(A_1 \cdot (A_2 \cdot A_3)) \cdot A_4$
3. $(A_1 \cdot A_2) \cdot (A_3 \cdot A_4)$.
4. $A_1 \cdot ((A_2 \cdot A_3) \cdot A_4)$
5. $A_1 \, (A_2 \cdot (A_3 \cdot A_4))$.

Ans 6.

Divide and Conque Approach	Dynamic Programming	Greedy Approach.
i). It deals with 3 steps of each level of recursion	1). It involves the sequence of four steps	1). In Greedy Approach, the optimal solution is optinied from a set of feasible solution.
a) Divide the problem into a number of subproblem	a) Charactrised the structure of optimal solution	
b). Conquer the sub-problem by solving then recusively	b). Recursively defines the value of optimal solution	
c). Combine the sol-ution to the sub problem into the solution for original subproblem	c). Compute the value of optimal solution in abottom up menimum	
	d). Construcy on optimal solution from compute information.	
2. It is recursive.	2). It is non-recursive.	2) It is iterative.
3. Top down approach	2). Bottom-up approch.	3). Top down approach.

4). In this subproblem are independent of each other.	4). In this sub problem are interdependent.	4). It contains a particular set of feasible set of solution.
ex: merge sort, Binary Search	Ex. MCM, 0/1 knapsack problem.	Ex. Fractional knapsack Problem, Activity selection problem.

Ans 2. Floyd-warshall's algorithm follows dynamic programming approach because the all pair shortest path are computed in bottom up manner.

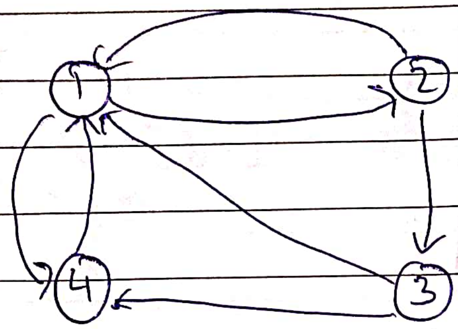
- The Floyd warshall Algorithm is used for solving the All pair shortest path problem. It finds shortest distance between every pair of vertices in a given edge weighted directed Graph.
- Algorithm of floyd warshall's is:-

```
Floyd ( W [1... n, 1.... n])
// Implement Floyd's algorithm for all pairs
   shortest path problem.
// Input : The weight matrix W of a graph with
   no negative length cycle.
// Output :- The distance matrix of the shortest
   paths lengths.
   D ← W // is not necessary if W can over
   written for k ← 1 to n do
        for i ← 1 to n do
   D[i, j] ← min { D[i, j], D[i, k]+ D[k, j]}
   return D
```

MIT SCHOOL OF ENGINEERING
Rajbaug, Loni-Kalbhor, Pune

UNIVERSITY
PUNE, INDIA
A leap towards World Class Education

8.



$$A_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & \infty \\ 5 & \infty & 0 & 1 \\ 2 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

$$A_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & \infty & 0 \end{bmatrix} \end{array}$$

$A^1 = [2,3] = 2$
$A^0 = [2,1] + [1,3]$
$= 8 + \infty$
$= \infty$
$2 < \infty$

$A^1 = [2,4] = \infty$
$A^0 = [2,1] + [1,4]$
$8 + 7$
$A^0 = 15$
$15 < \infty$

$A^1 = [3,2] = \infty$
$A^0 = [3,1] + [1,2]$
$= 5 + 3$
$= 8$
$\infty > 8$

$A^1 = [3,4] = 1$
$A^0 = [3,1] + [1,4]$
$= 5 + 7$
$= 12$
$1 < 12$

$A^1 = [4,2] = \infty$
$A^0 = [4,1] + [1,2]$
$= 2 + 3$
$= 5$
$\infty > 5$

$A^1 = [4,3] = \infty$
$A^0 = [4,1] + [1,3]$
$2 + \infty$
$A^0 = \infty$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & & & & \\ 2 & & & & \\ 3 & & & & \\ 4 & & & & \end{array}$$

$$A^2 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & 5 & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{bmatrix} \end{array}$$

$A^2 = [1,3] = \infty$
$A^1 = [1,2] + [2,3]$
$= 3+2$
$= 5$
$\infty > 5$

$A^2 = [1,4] = 7$
$A^1 = [1,2] + [2,4]$
$3 + 15$
18
$7 < 18$

$A^2 = [3,1] = 5.$
$A^1 = [3,2] + [2,1]$
$8 + 8$
16
$5 < 16$

$A^2 = [3,4] = 1$
$A^1 = [3,2] + [2,4]$
$= 8 + 15$
$= 23$
$1 < 23$

$A^2 = [4,1] = 2$
$A^1 = [4,2] + [2,1]$
$= 5 + 8$
$= 13$
$2 < 13$

$A^2 = [4,3] = \infty$
$A^1 = [4,2] + [2,3]$
$= 5 + 2$
$= 7$
$\infty > 7$

$A^3 = [1,4] = 7$
$A^2 = [1,3] + [3,4]$
$A^2 = 5 + 1$
$= 6$

$A^3 = [2,4] = 15$
$A^2 = [2,3] + [3,4]$
$= 2 + 1$
$= 3$
$15 > 3$

$A^3 = [1,2] = 3$
$A^2 = [1,3] + [3,2]$
$= 5 + 8$
$= 13$
$3 < 13$

$$A^4 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & 5 & 6 \\ 5 & 0 & 2 & 3 \\ 3 & 6 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{bmatrix} \end{array}$$

$A^4 = [1, 2] = 3$
$A^3 = [1,3] + [3,2]$
$= 5+8$
$= 13$

$3 < 13$

$A^4 = [3,1] = 5$
$A^3 = [3,4] + [4,1]$
$= 1+2 = 3$

$5 < 3$

$A^4 = [3,2] = 8$
$A^3 = [3,4] + [4,2]$
$= 1+8+6$

$8 > 6$

$A^4 = [2,1] = 7$
$A^3 = [2,3] + [3,1] = 2+3 = 5$, $7 > 5$

Ans10 The 0/1 knapsack problem means that the items are either completly or no item are filled in a knapsack.

ex: weights {3, 4, 6, 5}   wt of knapsack = 8 kg
Profits = {2, 3, 1, 4}   No of item = 4.

Step1. We write the weights & in ascending order and profits according to their wets:
wt = {3, 4, 5, 6}
P_i = {2, 3, 4, 1}

Step 2:

•	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0								
2	0								
3	0								
4	0								

Step 3.

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	2	2	2	2	2
2	0								
3	0								
4	0								

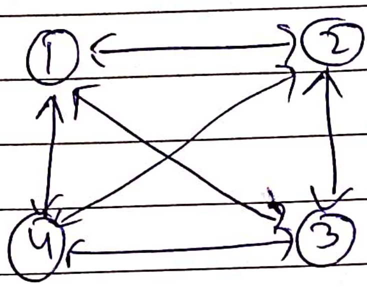
Step 5

	0	1	2	3	4	5	6	7	8	Wi	Pi
0	0	0	0	0	0	0	0	0	0	3	2
1	0	0	0	2	2	2	2	2	2	4	3
2	0	0	0	2	3	3	3	5	5	5	4
3	0	0	0	2	3	4	5	5	6	6	1.
4	0	0	0	2	3	4	5	5	6.		

$x_i = \{x_{i_1} \cdot x_2 \; x_3 \; x_4\}$

$x_i = \{1, 0, 1, 0\}$

Ans. 9: ~~8~~ TSP using DP.



$$A = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 15 & 20 \\ 2 & 5 & 0 & 9 & 10 \\ 3 & 6 & 13 & 0 & 12 \\ 4 & 8 & 8 & 9 & 0 \end{array}$$

Formula: $g(i, S) = \min\limits_{k \in S} \{ C_{ik} + g(k, S - \{k\}) \}$

where $g(i, S)$ is a cost of between i^{th} to S^{th} city.
C_{ik} is cost of moving from i to k city

$g(2, \phi) = 5$
$g(3, \phi) = 6$
$g(4, \phi) = 8$
$g(2, \{3\}) = 15$
$g(2, \{4\}) = 8$
$g(3, \{2\}) = 5$
$g(3, \{4\}) = 8$
$g(4, \{2\}) = 5$
$g(4, \{3\}) = 6$

$g(2, \{3, 4\}) = 25$
$g(3, \{2, 4\}) = 25$
$g(4, \{2, 3\}) = 23$

$\therefore g(1, \{2, 3, 4\}) = \min \{ C_{12} + g(2, \{3, 4\}), C_{13} + g(3, \{4\}), \}$

$= \min \{10 + 25, 15 + 25, 20 + 23\}$

$\min \{35, 40, 43\}$

$g(1, \{2, 3, 4\}) = 35$

Path is $1 \to 2 \to 3 \to 4$