NAME: PRERNA SUNIL JADHAV

SAP ID: 60004220127

AOA ASSIGNMENT 02

Q1 Find an optimal sol^n for following 0/1 knapsack problem using dp.

No. of objects = 4 = (n)

Capacity (m) = 5

weights ⇒

w1	w2	w3	w4
2	3	4	5

Profits ⇒

P1	P2	P3	P4
3	4	5	6

→

P_n	w_n	n \ m	0	1	2	3	4	5
		0	0	0	0	0	0	0
3	2	1	0	0	03	03	03	03
4	3	2	0	0~~3~~	03	04	04	07
5	4	3	0	0	03	04	05	07
6	5	4	0	0	03	04	05	07

$$KS(n, m) = \max\begin{cases} KS(n-1, m-w_n) + P_n \\ KS(n-1, m) \end{cases}$$

$$= 0 \quad \text{if} \quad n = 0 \quad \text{or} \quad m = 0$$

$$= KS(n-1, m) \quad \text{if} \quad w_n > m$$

⇒ $KS(1,1) = KS(0,1) \quad \because 2 > 1 \quad (w_n > m)$

$= 0$

$KS(1,2) = \cancel{KS\{}\ \max\begin{cases} KS(0,0)+3 \\ KS(0,2) \end{cases} = \max\begin{cases} 03 \\ 0 \end{cases} = 03$

$$KS(1,3) = \max \begin{cases} KS(0,1) + 3 \\ KS(0,3) \end{cases} = \max \begin{cases} 3 \\ 0 \end{cases} = 3$$

$$KS(1,4) = \max \begin{cases} KS(0,2) + 3 \\ KS(0,4) \end{cases} = \max \begin{cases} 3 \\ 0 \end{cases} = 3$$

$$KS(1,5) = \max \begin{cases} KS(0,3) + 3 \\ KS(0,5) \end{cases} = \max \begin{cases} 3 \\ 0 \end{cases} = 3$$

$(w_2 = 3)$
$(P_2 = 4)$

$$KS(2,1) = \cancel{\max} \; KS(1,1) \quad \because w_n > m$$
$$= 0$$

$$KS(2,2) = KS(1,2) \quad \because w_n > m$$
$$= 3$$

$$KS(2,3) = \max \begin{cases} KS(1,0) + 4 \\ KS(1,3) \end{cases} = \max \begin{cases} 4 \\ 3 \end{cases} = 4$$

$$KS(2,4) = \max \begin{cases} KS(1,1) + 4 \\ KS(1,4) \end{cases} = \max \begin{cases} 4 \\ 3 \end{cases} = 4$$

$$KS(2,5) = \max \begin{cases} KS(1,2) + 4 \\ KS(1,5) \end{cases} = \max \begin{cases} 3+4 \\ 3 \end{cases} = 7$$

$m = 1$
$w_n = 4$
$P_n = 5$

$$KS(3,1) = KS(2,1) \quad \because w_n > m$$
$$= 0$$

$$KS(3,2) = KS(2,2) \quad \because w_n > m$$
$$= 3$$

$KS(3,3) = KS(2,3) \quad \because w_n > m$
$= 4$

$KS(3,4) = \max \begin{cases} KS(2,0) + 5 \\ KS(2,4) \end{cases} = \max \begin{cases} 5 \\ 4 \end{cases} = 5$

$KS(3,5) = \max \begin{cases} KS(2,1) + 5 \\ KS(2,5) \end{cases} = \max \begin{cases} 5 \\ 7 \end{cases} = 7$

$w_4 = 5 \leftarrow KS(4,1) = \cancel{\max\{KS} = KS(3,1) \quad \because w_n > m$
$= 0$

$KS(4,2) = KS(3,2) \quad \because w_n > m$
$= 3$

$KS(4,3) = KS(3,3) \quad \because w_n > m$
$= 4$

$KS(4,4) = KS(3,4) \quad \because w_n > m$
$= 5$

$KS(4,5) = \max \begin{cases} KS(3,0) + 6 \\ KS(3,5) \end{cases} = \max \begin{cases} 6 \\ 7 \end{cases} = \underline{\underline{7}}$

~~KS~~ Profit $= 7$ can be achieved by placing object 2 $\therefore 7 - 4 = 3$
Now check if 3 is present in above row i.e., obj 1, yes it is check for the row above it, no its not so place obj 1

$\therefore 7 = \underset{\uparrow p_2}{4} + \underset{\uparrow p_1}{3} \Rightarrow$ Total profit

Q2

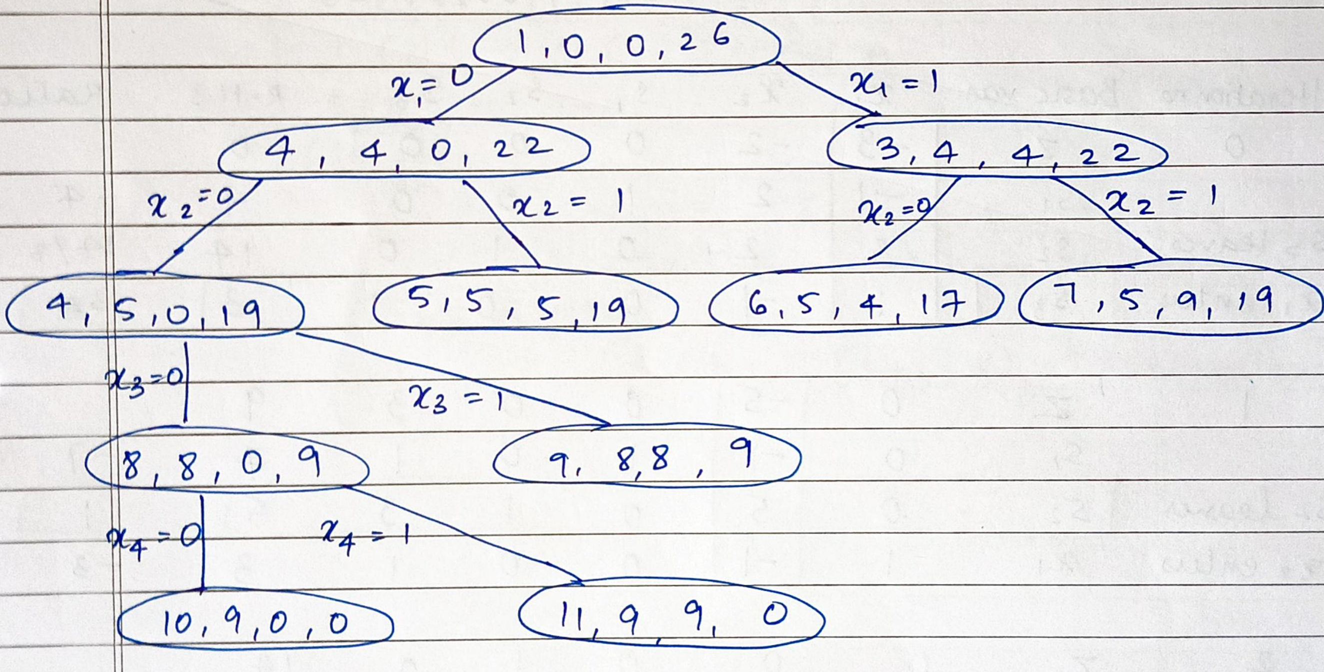
	Greedy Algo	Divide & conquer	Dynamic Programming
01.	follows Top-down approach	Follow Top-down approach	follows bottom-up approach
02.	solves the optimization problem	solves decision problem	solves the optimization problem
03.	Iterative in nature	Recursive in nature	Recursive in nature
04.	Efficient & faster than divide & conquer	Less efficient and slower	More Effecient and slower than greedy
05.	Extra memory is not required	some memory is required	more memory is required
06.	Eg: Dijkstra's Algo, knapsack (Fractional)	Eg: Merge sort, Quick sort, Binary search	Eg: 0/1 knapsack, Bellman ford, All pair shortest path

Q3 Write an algorithm for sum of subset & solve the given problem using written algorithm:

$N = 4$, $w = \{4, 5, 8, 9\}$, required sum $= 9$

Algorithm:

```
sum of subset (s, k, r)
    x[k] = 1
    if (s + w(k] = m):
        print (x[1:m])
    else if (s + w[k] + w[k+1] ≤ m)
        sumofsubset (s+ w[k], k+1 , r-w[k])
    if [(s+r - w[k] ≥ m) & (s+w [k+1] ≤ m)]
        x[k] = 0
        sumqset (s, k+1, r-w[k]);
```



Possible solution

[0, 0, 0, 1], [1, 1, 0, 0]

Q4 using simplex method determine value of z

Maximize $z = 3x_1 + 2x_2$

subject to
$$-x_1 + 2x_2 \leq 4$$
$$3x_1 + 2x_2 \leq 14$$
$$x_1 - x_2 \leq 3$$
$$x_1, x_2 \geq 0$$

$\rightarrow$

$$z - 3x_1 - 2x_2 + 0s_1 + 0s_2 + 0s_3 = 0$$
$$-x_1 + 2x_2 + s_1 + 0s_2 + 0s_3 = 4$$
$$3x_1 + 2x_2 + 0s_1 + s_2 + 0s_3 = 14$$
$$x_1 - x_2 + 0s_1 + 0s_2 + s_3 = 3$$

Iteration no	Basic var.	x_1	x_2	s_1	s_2	s_3	R.H.S	Ratio
0	z	-3	-2	0	0	0	0	
	s_1	-1	2	1	0	0	4	-4
s_3 leaves	s_2	3	2	0	1	0	14	14/3
x_1 enters	s_3	1	-1	0	0	1	3	3
1	z	0	-5	0	0	3	9	
	s_1	0	-1	1	0	1	7	-7
s_2 leaves	s_2	0	5	0	1	-3	5	1
x_2 enters	x_1	1	-1	0	0	1	3	-3
2	z	0	0	0	1	0	14	
	s_1	0	0	1	1/5	2/5	8	
	x_2	0	1	0	1/5	-3/5	1	
	x_1	1	0	0	1/5	2/5	4	

$\therefore x_1 = 4, \quad x_2 = 1$

Q5

a) Rabin Karp String matching Algorithm

n = T.length
m = P.length
h = d^{m-1} mod q
p = 0
d = 0
t = 0
for i = 1 to m:
    p = (d.p + P[i]) mod q
    t = (dt + T[i]) mod q
for s = 0 to n-m
    if p == ts
        if P[1...m] == T[s+1.... s+m]
            print "Pattern match with shift" s
    if s < n-m
        t = (d(t - T[s+1]h) + T[s+m+1] mod

b) Knuth Morris Pratt

```
Compute Prefix function (P)
    n = P. length
    Let π[1...m] be new array
    π [1] = 0
    K = 0
    for q = 2 to m
        while K > 0 && P[K+1] ≠ P[q]
                K = π[K]
        if P[K+1] == P[q]
                K = K+1
        π[q] = K
    return π

KMP (T, P)
  n = T. length
  m = P. length
  π = compute Prefix function (P)
  q = 0
  for i = 1 to n
      while q > 0 && P[q+1] ≠ T[i]
               q = π [q]
      if P [q+1] == T[i]
            q = q+1
      if q == m
            print (" Pattern occurs with shift",
                          i - m)
            q = π [q]
```

FOR EDUCATIONAL USE