Shreya Shah
60004210045
A1 comps
AOA

Assignment - 2

Q1] $w = \{2, 3, 4, 5\}$ $m = 5$
$P = \{3, 4, 5, 6\}$
$S^0 = \{(0,0)\}$
$S_1^0 = \{(3,2)\}$
$S^1 = \{(0,0), (3,2)\}$
$S_2^1 = \{(4,3), (7,5)\}$
$S^2 = \{(0,0), (3,2), (4,3), (7,5)\}$
$S_3^2 = \{(5,4), (8,6), \cancel{(9,7)}, \cancel{(12,9)}\}$
$S^3 = \{(0,0), (3,2), (4,3), (7,5), (5,4), \cancel{(8,8)}\}$
$S_4^3 = \{(6,5), \cancel{(9,7)}, \cancel{(10,8)}, \cancel{(13,10)}, \cancel{(11,9)}\}$
$S^4 = \{(0,0), (3,2), (4,3), (6,5), \boxed{(7,5)}, (5,4), (6,5)\}$
max profit

$(7,5) - (4,3) = (3,2)$ O_2
$(3,2) - (3,2) = 0$ O_1
$\boxed{[1, 1, 0, 0]}$

Q2] **Dynamic:**
(i) Guarantee optimal solution
(ii) Subproblems overlap
(iii) Does more work compared to both
(iv) No specialized set of feasible solutions
(v) Employ memorization

Divide and conquer:
(i) Does not aim for optimal solution
(ii) Subproblems are independent
(iii) Slower and inefficient
(iv) No memorization

Greedy:
(i) Does not guarantee optimal solution
(ii) Subproblems do not overlap
(iii) Does little work
(iv) Select choice which is locally optimum
(v) No memorization

Q3] Algorithm.

Sum_of_subsets (s, k, r)

{ // Find all subsets of w[1:n] that sum to m

x[k] = 1

if (s + w[k] == m)

write x[1 : k]

else if (s + w[k] + w[k+1] ≤ m)

sum of subsets(s + w[k], k+1, r - w[k])

if (s + r - w[k] ≥ m) and (s + w[k+1] ≤ m)

x[k] = 0

sum-of-subsets(s, k+1), r - w[k])

eg: n = 4, w = {4, 5, 8, 9} & m = 9, set s = 0

Step 1 : i = 1, Adding item w_i

~~sum = sum + wi~~ $s = s + w_i = 0 + 4 = 4$

$s \le m$, ∴ add to soln set

x[i] = x[1] = 1 ⇒ x = [1, 0, 0, 0]

Step 2 : i = 2, Adding item w_i

$s = s + w_i = 4 + 5 = 9$

∴ s == m

so solution found & add to sol^n set

x[2] = 1 ⇒ x = [1, 1, 0 0]

PTO

FOR EDUCATIONAL USE

Sundaram®

6000421004S
Shreya Shah

Q4] Max $Z - (3x_1 + 2x_2 + 0s_1 + 0s_2 + 0s_3) = 0$

subject to $-x_1 + 2x_2 + s_1 = 4$

$3x_1 + 2x_2 + s_2 = 14$

$x_1 - x_2 + s_3 = 3$

Simplex Table:

Iteration No.	Basic Variable	x_1	x_2	s_1	s_2	s_3	RHS Soln	Ratio
0	Z	-3	-2	0	0	0	0	
	s_1	-1	2	1	0	0	4	–
	s_2	3	2	0	1	0	14	$\frac{14}{3} = 4.67$
	s_3	1	-1	0	0	1	3	$\frac{3}{1} = 3$
1	Z	0	-5	0	0	3	9	
s_3 exits	s_1	0	1	1	0	1	7	7
x_1 enters	s_2	0	5	0	1	-3	5	1
	x_1	1	-1	0	0	1	3	–
1	Z	0	0	0	1	0	14	
s_2 leaves	s_1	0	0	1	-1/5	8/5	6	
x_2 enters	x_2	0	1	0	1/5	-3/5	1	
	x_1	1	0	0	1/5	2/5	4	

$x_1 = 4, x_2 = 1, Z_{max} = \boxed{14}$

Q5] (a) RabinKarp Matcher (T, P, d, q)

```
n = T.length
m = P.length
h = d^(m-1) mod q
p = 0
t0 = 0
for i = 1 to m
    p = (dp + P[i]) mod q
    t0 = (dt0 + T[i]) mod q
for s = 0 to n-m
    if p == ts
        if P[1...m] == T[s+1 ... s+m]
            print ("Pattern occurs with shift" s)
    if s < n-m
        ts+1 = (d(ts - T[s+1] h) + T[s+m+1]) mod q
```

(b) Compute Prefix Function (P)

```
m = P.length
let π[1...m] be a 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-matcher (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] ≠ P[i]
        q = π[q]
    if P[q+1] == T[i]
        q = q+1
    if q == m
        Print ("Pattern occurs with shift" i)
    q = π[q]
```