Name : Solanki Harshbhai
DIV : SE4-D
ROLL-NO : 62

Page No.
Date 11 5 21

Assignment - 1

Q-1 Define Big-O, Θ, Ω notation, find the complexity of given recurrence relation using master method.

(i) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

(ii) $T(n) = 4T\left(\frac{n}{4}\right) + n^3$

Ans * **Big Oh (O)** :- Let f(n) and g(n) are two non-negative function indicating running time of two algorithm. We say, g(n) is upper bound of f(n) if there exist some positive constan c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$. It denoted as $f(n) = O(g(n))$

* **Big omega (Ω)** :- Let f(n) and g(n) are two non-negative function indicating running time of two algorithm. We say the function g(n) is lower bound function f(n) if there exist some positive constant c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$. It is denoted as $f(n) = \Omega(g(n))$

* **Big Theta (Θ)** :- Let f(n) and g(n) are two non-negative function indicating running time of two algorithms. We say the function g(n) is tight bound of function f(n) if there exist some positive constants c_1, c_2 and n_0 such that $0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$. It is denoted as $f(n) = \Theta(g(n))$

(i) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

$\Rightarrow$ Given equation is in form of $T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$

Here, $a = 4$, $b = 2$, $f(n) = n^2$, $d = 2$.

$b^d = 2^2 = 4$

From master theorem $a = b^d$ so

$T(n) = \theta(n^d \log n) = \boxed{\theta(n^2 \log n)}$

(ii) $T(n) = 4T\left(\frac{n}{4}\right) + n^3$

$\Rightarrow$ Given equation is in form of $T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$

Here $a = 4$, $b = 4$, $f(n) = n^3$, $d = 3$

$b^d = 4^3 = 64$

From master theorem $a < b^d$

$\therefore T(n) = \theta(n^d)$

$\boxed{T(n) = \theta(n^3)}$

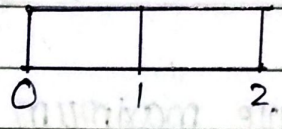
Page No.
Date

Q2 Let $n=4$ $(P_1, P_2, P_3, P_4) = (25, 2, 4, 7)$ and $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$ find feasible solution using job sequencing with deadline. Also find optimal solution using job sequencing with deadlines.

⇒

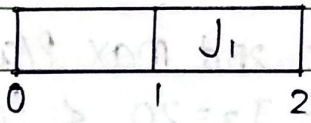
Job	J_1	J_2	J_3	J_4
Profit	25	2	4	7
deadline	2	1	2	1

Initially;

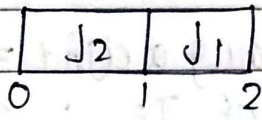


Iteration: 1

JOB J_1 Slot (1-2) Profit = 25



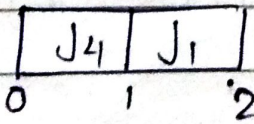
Iteration 2: JOB J_2 Slot (0-1) Profit = 2



Iteration 3: JOB J_3 deadline = 2 ∵ (1-2) is already allocated and Profit is less so discarded.

Iteration 4: JOB J_4 ∵ Profit of $J_4 > J_2$

J_4 Slot (0-1) Profit = 7.



∴ JOB sequence is $\{J_4, J_1\}$ and max Profit = 32

Q-3 Consider the following instance of objects where first value is Profit and second value is weight. Find max Profit using greedy method where knapsack size is 50

A = { (60, 10), (100, 20), (120, 30) }

⇒

item	Profit	weight	P/W
I_1	60	10	6
I_2	100	20	5
I_3	120	30	4

M = 50

Iteration-1: I_1 have maximum P/w ratio.
∵ weight of I_1 < M.
50 − 10 = 40 remaining

Iteration-2: I_2 have 2nd max P/w ratio.
∵ weight of I_2 = 20 < 40
∴ 40 − 20 = 20 remaining

Iteration-3: ∵ remaing weight = 20 which is less than weight of I_3 ∴ 20 is selected out of 30 ∴ 20 − 20 = 0

∴ Profit = $\left\{ 60 + 100 + \left(120 \times \frac{20}{30} \right) \right\}$

∴ Profit = 60 + 100 + 80 = 240

Page No.
Date

Q-4 Given a string T = "bacbababacaab" and a pattern P = "ababaca". use KMP algorithm to find whether P occurs in T

⇒ The π function of given pattern is:

i	1	2	3	4	5	6	7
P[i]	a	b	a	b	a	c	a
π[i]	0	0	1	2	3	0	1

Initially n = size of T = 15;
m = size of P = 7

Step 1: i = 1, q = 0
comparing P[1] with T[1]

T	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P	a	b	a	b	a	c	a.

P[1] doesn't match with T[1] so P will shifted one position to the right.

Step 2: i = 2, q = 0
comparing P[1] with T[2]

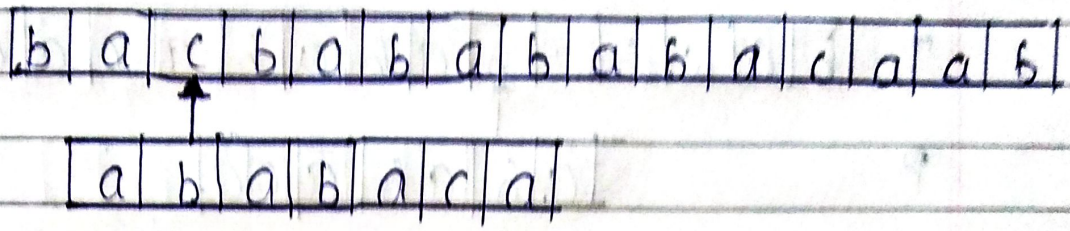
T	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P	a	b	a	b	a	c	a.

P[1] is match with T[2].

Step3: i = 3, q = 1

comparing P[2] with T[3]

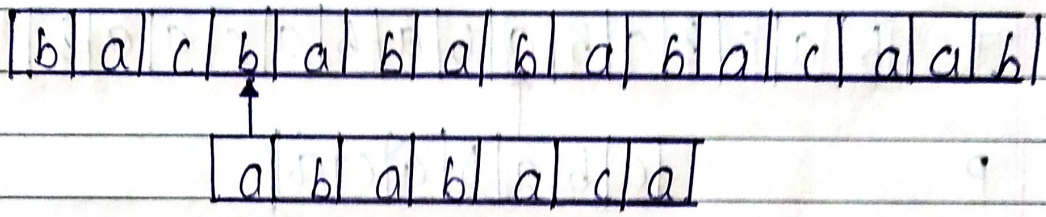


P[2] does not match with T[3]

Backtracking on P, comparing P[1] and T[3]

Step4: i = 4, q = 0

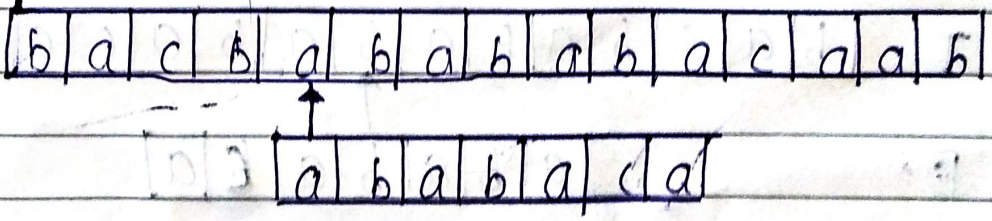
comparing P[1] with T[4]



P[1] does not match with T[4].

Step5: i = 1, q = 0

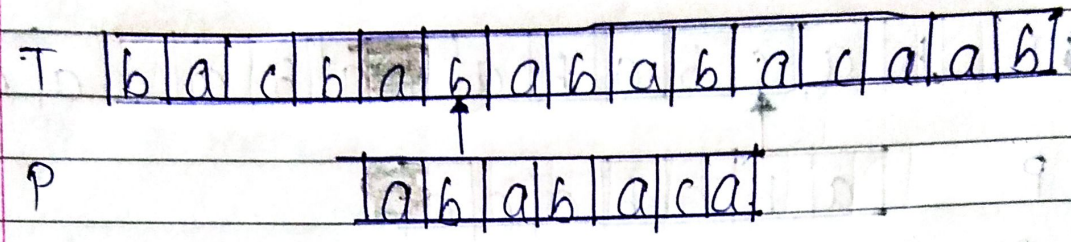
comparing P[1] with T[5]



P[1] matches with T[5].

Step 6: $i = 6$, $q = 1$

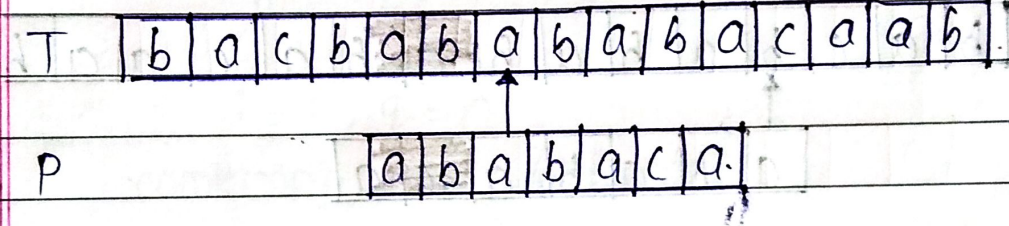
comparing P[2] with T[6]



P[2] matches with T[6]

Step 7: $i = 7$, $q = 2$

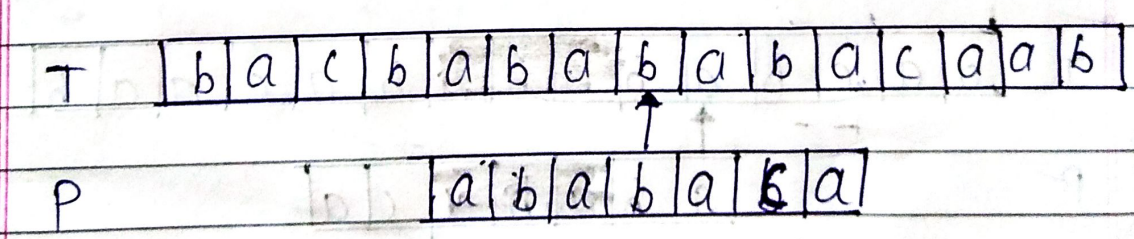
comparing P[3] with T[7]



P[3] Matches with T[7]

Step 8: $i = 8$ $q = 3$

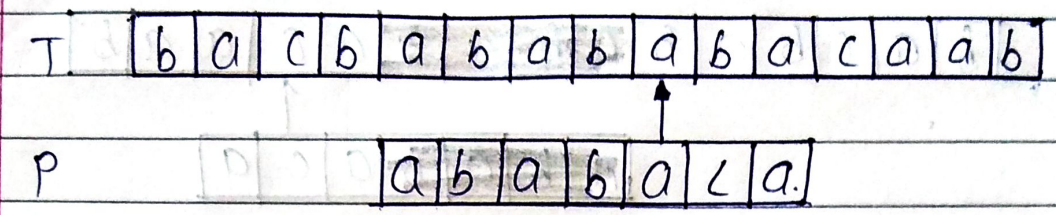
comparing P[4] with T[8]



P[4] matches with T[8]

Step 9: i = 9, q = 4

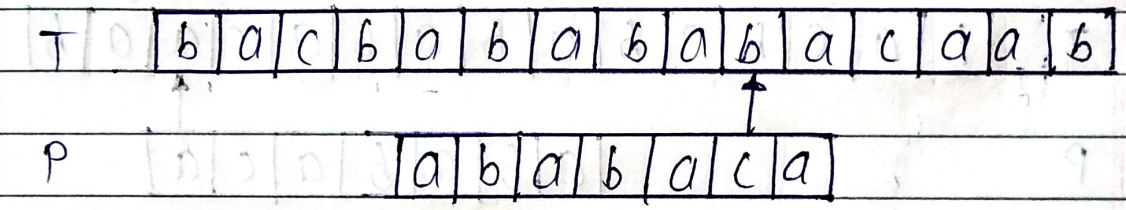
comparing P[5] with T[9]



P[5] matches with T[9]

Step 10: i = 10, q = 5

comparing P[6] with T[10]



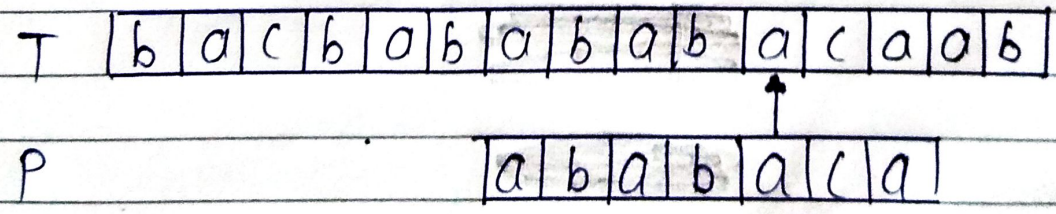
P[6] does not match with T[10]

Backtracking on P, comparing P[4] with T[10]

Because after miss match $\pi[5] = 3$

Step-11: i = 1, q = 4

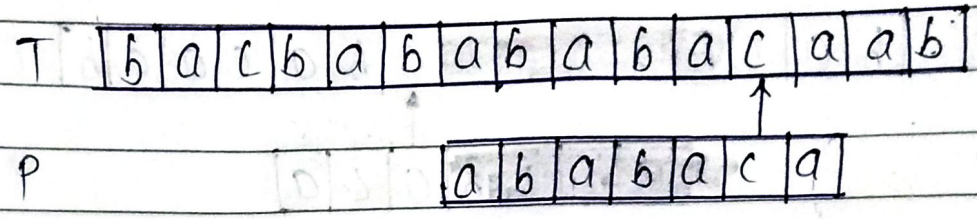
comparing P[5] with T[11]



P[5] matches with T[11]

Step-12: $i = 12, q = 5$

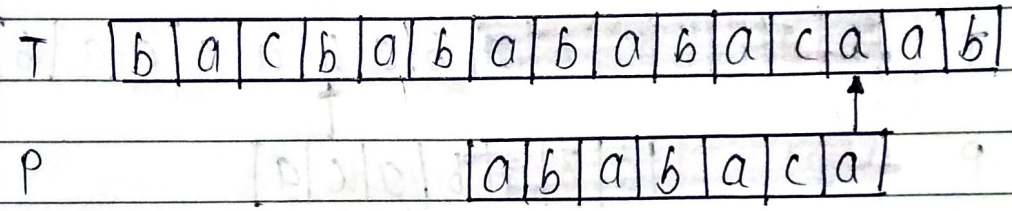
comparing P[6] with T[12]



P[6] match with T[12]

Step 13: $i = 13, q = 6$

comparing P[7] with T[13]



P[7] match with T[13]

⟹ Patter P is found to be occur in string T

Total number of shift = $i - m = 13 - 7 = 6$ shift