Explanations

BRAC UNIVERSITY

Abir Ahammed Bhuiyan
ID: 20101197
Dept. of Computer
Science and Engineering

Section : 04
Subject : CSE221
Semester : Summer-2022
Date : 23. 06. 2022

Answer to the Ques no: 2

Implementation 1:

$$T(n) = T(n-1) + T(n-2) + c.$$

assuming,

$$T(n-1) = T(n-2).$$

now,

$$T(n) = 2T(n-1) + c.$$

$$T(n-1) = 4T(n-2) + 3c.$$

$$T(n-2) = 8T(n-3) + 7c.$$

$$T(n-k) = 2^k T(n-k) + (2^k - 1)c.$$

assume,

$$T(n-k) = T(0)$$

$$n-k = 0.$$

$$\therefore n = k$$

$$\therefore T(n) = 2^n T(0) + (2^n - 1)c. \quad [\because T(0) = 1].$$

$$\therefore T(n) = (1+c)2^n - c$$

$\therefore T(n) = O(2^n)$. this is the worst complexity implementation 2:

this Program is just fetching existing value and adding them and appending in the list. All this operation is in constant time. Also, this will keep happening for n times.

$$\therefore T(n) = O(1) + O(1) + O(1) + \ldots \ldots$$

$$= n \cdot O(1).$$

$$\therefore T(n) = O(n).$$

worst time complexity, $T(n) = O(n)$.

Therefore, implementation-2 is better than implementation-1 because implementat -2 has $O(n)$ time complexity.

Answer to the Question 4

for any given input the main function that will calculate the result will iterate for $n \times n \times n$ or n^3 times.

$\therefore$ worst time complexity $O(n^3)$.

Ans to the Ques no: 5

$$T(n) = T(n/2) + n-1 \quad ; \quad T(1) = 0$$

$$\left.\begin{array}{c} n-1 \\ \downarrow \\ \frac{n}{2}-1 \\ \downarrow \\ \frac{n}{4}-1 \\ \vdots \\ \frac{n}{2^k}-1 \end{array}\right] \text{total } k \text{ steps}$$

$$\therefore T(n) = (n-1) + \left(\frac{n}{2}-1\right) + \left(\frac{n}{4}-1\right) + \dots + \left(\frac{n}{2^k}-1\right)$$

$$= \left[n + \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{2^k}\right] - \underbrace{\left[1+1+\dots+1\right]}_{\text{total } k \text{ times}}$$

$$= n\left[1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^k}\right] - k$$

assuming,

$$\frac{n}{2^k} = 1$$

$$\therefore k = \log_2 n$$

$$\therefore T(n) = n\left[1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^k}\right] - \log_2 n$$

$$= n \cdot 1 - \log_2 n \qquad \left[\because 1 + \frac{1}{2} + \frac{1}{4} + \cdots \approx 1\right].$$

$$= n - \log_2 n$$

$\therefore$ Worst case time complexity, $T(n) = O(n)$.

(2)

$T(n) = T(n-1) + n-1 \quad , \quad T(1) = 0.$

$$n-1 \downarrow n-2 \downarrow n-3 \cdots 3 \downarrow 2 \downarrow 1$$

$\therefore T(n) = (n-1) + (n-2) + (n-3) + \dots + 3 + 2 + 1$

Here,

$$d = (n-2) - (n-1) = n - 2 - n + 1 = -1$$

$$a = n-1$$

Arithmetic series summation,

$$S_n = \frac{n}{2}(2a + (n-1)d)$$

$$= \frac{n}{2}[2(n-1) + (n-1)(-1)]$$

$$= \frac{n}{2}[2n - 2 - n + 1]$$

$$= \frac{n}{2}[n-1]$$

$$= \frac{1}{2}(n^2 - n)$$

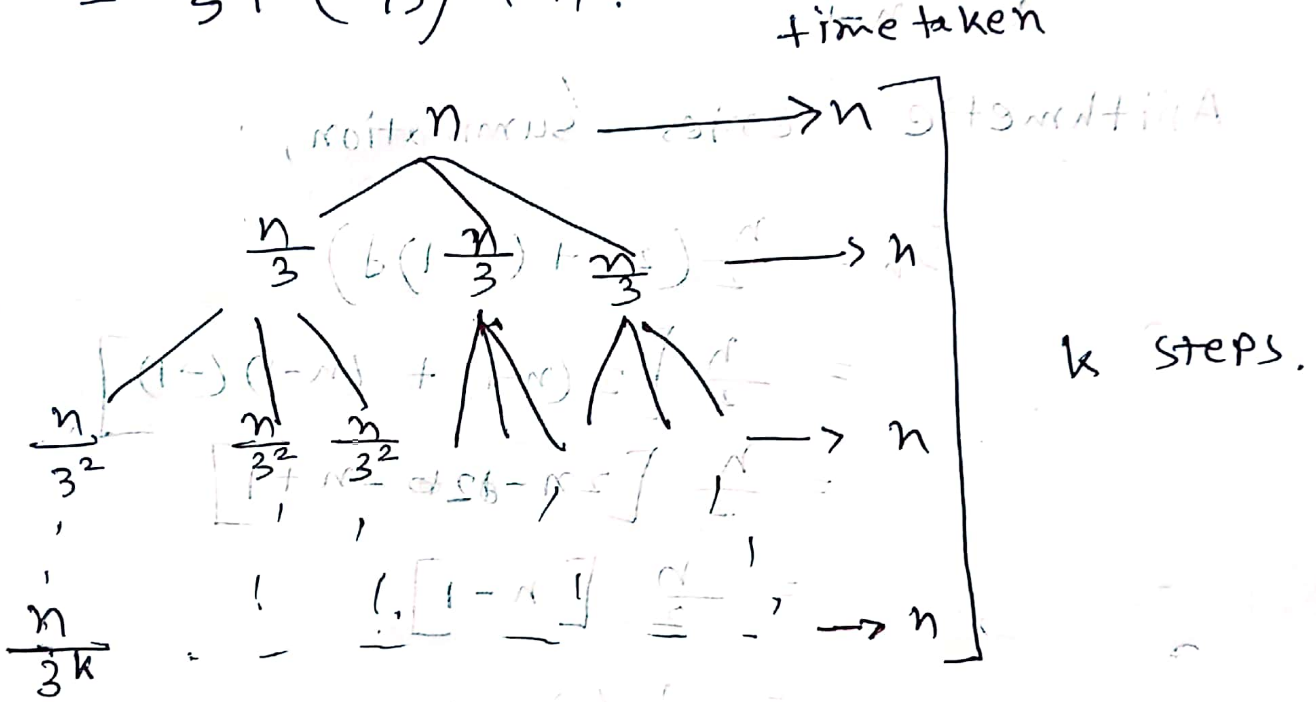
$$\therefore T(n) = n^2 - n$$

Worst case time complexity, $T(n) = O(n^2)$

(3)

$$T(n) = T(n/3) + 2T(n/3) + n$$

$$= 3T(n/3) + n.$$



$\therefore T(n) = n\ k$

assuming,

$$\frac{n}{3^k} = 1$$

$$n = 3^k$$

$$\therefore k = \log_3 n.$$

$\therefore T(n) = n \log_3 n$

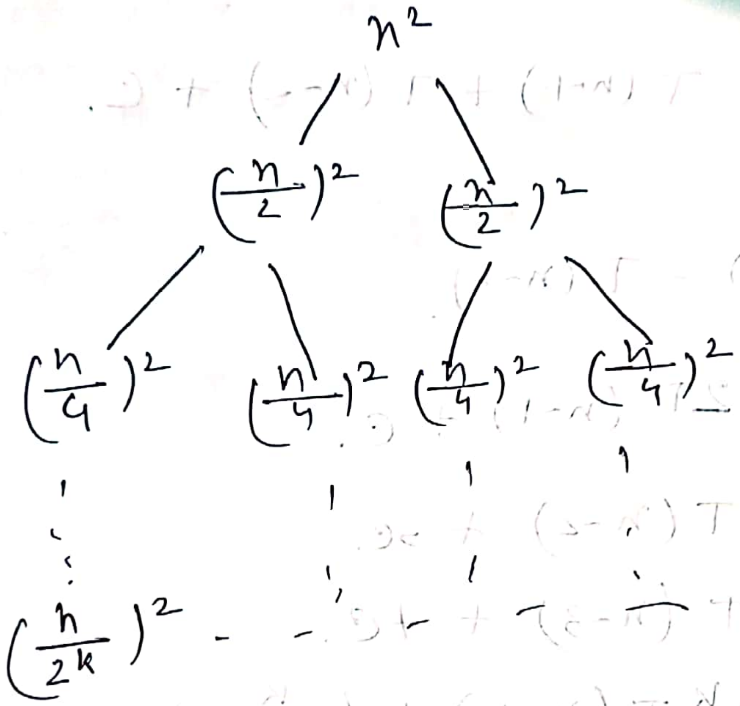
$\therefore$ worst time complexity, $T(n) = O(n \log_3 n)$

(4)

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

time taken

n^2



$2.\left(\frac{n}{2}\right)^2$

$2^2\left(\frac{n}{4}\right)^2$

$2^k\left(\frac{n}{2^k}\right)^2$

$$\therefore T(n) = n^2 + 2.\left(\frac{n}{2}\right)^2 + 2^2\left(\frac{n}{4}\right)^2 + \cdots\cdots + 2^k\left(\frac{n}{2^k}\right)^2$$

$$= n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \cdots + \frac{n^2}{2^k}$$

$$= n^2\left[1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^k}\right]$$

$$= n^2 . 1 \qquad \left[\because 1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^k} \approx 1\right].$$

$$= n^2$$

$\therefore$ worst case time complexity, $T(n) = O(n^2)$

CamScanner