02/08/2022

Tutorial - 2

Piku Maity
1RV21MC071

1. Find the time Complexity of an algorithm which has two executable ~~Path~~. Part.

(a) Sorting Part that takes - $\frac{1}{2}n(n-1)$ Comparisons.

(b) To check Consecutive element, it requires n Comparison.

Sol^n ->

(a) With the help of Property of asymptotic notation, order of growth for Sorting Path is

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

$$= \frac{n^2-n}{2}$$

∴ It is approximately ($\approx$) n^2

∴ Quadratic in Nature.

(b) order of growth checking Consecutive element is n Comparison which is approximately ($\approx$) n.

∴ Linear in Nature

∴ Let, $f_1(n) = n^2$

& $f_2(n) = n$

$f(n) = f_1(n) + f_2(n) = \max\{n^2, n\}$

$f_2(n) \approx n^2$

② Order the following functions according to their order of growth from the lowest to the highest.

$(n-2)!\ ,\ 5\log(n+100)^{10},\ 2^{2n},\ 10.$

∴ order of growth $(n-2)! \rightarrow \overset{(n-2)!}{\cancel{n!}}$ (Factorial in nature).

order of growth $5\log(n+100)^{10} \rightarrow \log(n)^{10}$ (logarithmic)

order of growth $2^{2n} \rightarrow 2^{2n}$ (Exponential in nature)

order of growth $10 \rightarrow 10$ (constant in nature)

∴ According to order of growth from the lowest to the highest is $10 \leq 5\log(n+100)^{10} \leq 2^{2n} \leq (n-2)!$

③ Solve the following recurrence relation

$$X(n) = x(n/3) + 1 \text{ for } n>1,\ x(1)=1$$

$$\therefore X(n) = \begin{cases} 1 & \text{if } n=1 \rightarrow \text{Initial Condition.} \\ x(n/3)+1 & \text{Otherwise.} \end{cases}$$

∴ Consider $x(n) = x(n/3)+1$

So, $x(n) = [x(n/9)+1]+1$

$x(n) = [x(n/27)+1]+2$

$x(n) = x(n/3)+1$ —①

Replace n by n/3

$x(n/3) = x\left(\frac{n/3}{3}\right)+1$

$x(n/3) = x(n/9)+1$

∴ Replace n by n/3

$x\left(\frac{n/3}{3}\right) = x\left(\frac{n/3}{9}\right)+1$

$x(n/9) = x(n/27)+1$

$$\therefore x(n) = \left[x\left(\frac{n}{27}\right)+1\right]+2$$

$$= x\left(\frac{n}{3^{i}}+3\right)$$

$$= x\left(\frac{n}{3^{i}}+i\right).$$

④ Evaluate the given Postfix Expression.

53 + 6 2 / * 3 5 * +

Scanned character	Operand Stack	Operation
5	5 → TOP	
3	Pop { 3 → TOP 5	
+	8	OP2 = 3, OP1 = 5 5+3 = 8
6	6 → TOP 8	
2	2 → TOP 6 8	OP2 = 2, OP1 = 6 6 / 2 = 3
/	3 8	
*	24	OP2=3, OP1=8

3	3 24	
5	5 3 24	
*	15 → top 24	OP2=5, OP1=3 3*5=15
+	39	OP2=15, OP1=24 24+15=39

· No character left out in Postfix. expⁿ. So, the top of the stack is the result. So, 39.

∴) Implement the Dynamic memory allocation using Stack.