Set A

Ans to the qus no .(1)

A **recurrence** is an equation or inequality that describes a function in terms of its value on smaller inputs.

properties of Recursion:

- Must have a base value where it terminates
- The recursive call should progress in such way that every time a recursive call comes closer to the base criteria.

  some conventional techniques to analyze recursion Algorithms.

- **Repeated (backward) substitution method**
  - Expanding the recurrence by substitution and noticing a pattern (this is not a strictly formal proof).
- **Substitution method**
  - guessing the solutions
  - verifying the solution by the mathematical induction
- **Recursion-trees**
- **Master method**
  - templates for different classes of recurrences

Set A

Ans to the qus no .(2)

Best Case	**Average Case**	**Worst Case**
Provides a lower bound on running time	Provides a prediction about the running time	Provides an upper bound on running time
Input is the one for which the algorithm runs faster	Assumes the input is random	An absolute guarantee that the algorithm run longer ,no matter what the inputs

Ans to the qus no .(3)

T(n)= $\begin{Bmatrix} 1, n = 1 \\ 3T\left(\frac{n}{3}\right) + n, n > 1 \end{Bmatrix}$

repeated backward substitution method:

T(n)= $\begin{Bmatrix} 1, n = 1 \\ 3T\left(\frac{n}{3}\right) + n, n > 1 \end{Bmatrix}$

Set A

$T\left(\frac{n}{3}\right)=3T\left(\frac{n}{3^2}\right)+\frac{n}{3}$

$T\left(\frac{n}{3^2}\right)=3T\left(\frac{n}{3^3}\right)+\frac{n}{3^2}$

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

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

$=3\ \{\ 3T\left(\frac{n}{3^2}\right)+\frac{n}{3}\ \}+n$

$=3^2\ \{\ T\left(\frac{n}{3^2}\right)\ \}+n+n$

$=3^2\ \{\ T\left(\frac{n}{3^2}\right)\ \}+2n$

$=3^2\ \{\ 3T\left(\frac{n}{3^3}\right)+\frac{n}{3^2}\ \}+2n$

$=3^3\ T\left(\frac{n}{3^3}\right)+n+2n$

$=2^3\ T\left(\frac{n}{3^3}\right)+3n$

$=3^3\ \{\ 2T\left(\frac{n}{3^4}\right)+\frac{n}{3^3}\}+3n$

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

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

After substituting k-1 times, we get,

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

So, $T(n)=2^k\ T\left(\frac{n}{2^k}\right)\ +kn$

Replace k by log n,we get,

So,$\frac{n}{2^k}$=1 =>$2^k = n$ =>log2^k=logn =>klog2=logn =>k=logn.

T(n)= $2^k \ T\left(\frac{n}{2^k}\right) + kn$ replace,

T(n)=n T(1)+(log n)*n

=n*1+n log n

=n+n log n

Therefore,T(n)=n+ n log n

Big O

T(n)=n+ n log n

T(n)<n log n+n log n

T(n)<2n log n, where ,n>=2

C=2,g(n)=n log n,n0=2

So,T(n) is O(n log n)

Big Ω

T(n)=n+ n log n

T(n)>n log n, where , n>=2

C=1 ,g(n)=n log n,n0=2

So,T(n) is Ω(n log n)

Big Θ

n log n<T(n)<2n log n ,n>=2

c1=1,c2=2,g(n)=n log n,n0=2

so,T(n) is *Θ(n log n)*

Master method :

T(n)= $3T\left(\frac{n}{3}\right)+n,\ n>1$

a=3,b=3,f(n)=n

Step 1: $n^{log_b a}$=$n^{log_3 3}$=n

Step 2: compare f(n) and $n^{log_b a}$------> case 2

Step 3: T(n) is Θ($n^{\log 3}\log n$)

--> T(n) is Θ(n $\log n$)

Set A

Ans to the qus no.(4)

```
Int insertionSort(int arr[],int n)
{
        int i, key, j;                      ----------------------------→ 0
        for(i = 1; i < n; i++)              ----------------------------→ 1+n+(n-1)=2n
       {
            key = arr[i];                   ----------------------------→ 2*(n-1)=2n-2
            j = i - 1;                      ----------------------------→ 2(n-1)=2n-2
            while(j >= 0 && arr[j] > key) ------------------------→x+ 3y
           {
                arr[j + 1] = arr[j];        --------------------------→4y
                j = j - 1;                  --------------------------→y
           }
           arr[j + 1] = key;                ----------------------------→3*(n-1)=3n-3
       }
}
```

x=2+3+……….+n= n(n+1)/2-1

y=1+2+3+……….+n-1= n(n-1)/2

Set A

$T(n)=0+2n+2n-2+2n-2+x+ 3y+4y+y+3n-3$

$=2n+2n-2+2n-2+x+3y+4y+y+3n-3$

$=9n-7+x+8y$

$=9n-7+ \{n(n+1)/2-1\} +8\{n(n-1)/2\}$

$=9n-7+0.5n^2+0.5n-1+4n^2-4n$

$=4.5n^2+5.5n-8$

Big o:

$T(n)= 4.5n^2+5.5n-8$

$T(n)< 4.5n^2+5.5n^2$

$T(n)< 10n^2,\ n>=1$

Where, $c=10$, $g(n)=n^2$, $n0=1$

so $T(n)$ is $O(n^2)$.

Big Omega :

$T(n)= 4.5n^2+5.5n-8$

$T(n)>4.5n^2-2.5n^2$

$T(n)>2n^2,\ n>=1$

Where, $c=2$, $g(n)= n^2$, $n0= 1$

So, $T(n)$ is $\Omega(n^2)$

Big Theta:

$2n^2 < T(n) < 10n^2$, $n >= 1$

Where, $c_1 = 2$, $c_2 = 10$, $g(n) = n^2$, $n_0 = 1$

So , $T(n)$ is $\Theta(n^2)$

Space complexity:

List of variables: i,j,key,n,arr

4n[arr]+4 [n]+4 [j]+4 [i]+4[key] = 4n+16

Ans to the qus no.(5)

ID is 19-40903-2-2 then,

D1=4, D2=1, D3=9, D4=1, D5=3

A = {7, D1, 3, D2, 2, D3, 6, D4, 2, 3, D5}

={7,4,3,1,2,9,6,1,2,3,3}

Set A

A =

index	0	1	2	3	4	5	6	7	8	9	10
number	7	4	3	1	2	9	6	1	2	3	3

max=9

min=1

Count =

index	0	1	2	3	4	5	6	7	8	9
number	0	0	0	0	0	0	0	0	0	0

Count=

index	0	1	2	3	4	5	6	7	8	9
number	0	1,1	1,1	1,1,1	1	0	1	1	0	1
	0	2	2	3	1	0	1	1	0	1

Cumulative Count =

index	0	1	2	3	4	5	6	7	8	9
number	0	2	4	7	8	8	9	10	10	11
(n-1)		1,0	3,2	6,5,4	7		8	9		10

Output=

index	0	1	2	3	4	5	6	7	8	9	10
number	1	1	2	2	3	3	3	4	6	7	9