15/02/23 Wednesday.

Q:- Write a program to display sum of first 50 natural numbers?

Sol:-
```
#include<iostream>
using namespace std;
int main()
{
    int sum=0;
    for(int i=0; i<=50; i++)
    {   sum= sum+i;
    }   cout << sum\n;
}
```

=> for 500 numbers:-

```
{
          1(1)
    int   sum = 0;

              1(1)      1(500)    2(500)
    for (int  i=1;      i<=500;   i++)

                 2(500)
       sum =  sum + i;

              (1)
       cout << sum;
}
```

=> Counting Basic operations

∴ 1+1+500+1000+1000+1

= 2503 => Complexity.

=> for "n" numbers:-

```
          1(1)
{  int  sum=0;
       1(1)    1(n)      2(n)
for(i=1;      i<=n ;     i++)
              2(n)
{   sum=sum+ i;
              (1)
}  cout <<sum;
}
```

∴ 1+1+n+2n+2n+1

= 5n+3 => Complexity.

∵ Efficient -> Time

Emperical Analysis
- (cost + time required)
- (not suitable solution)
- (Implement both solutions)

Analytical Analysis
- i) Basic operations
- ii) frequency of basic operations (B.O)
- iii) Sum of B.O.
- iv) Map on standard function.

16/02/23 Thursday

for Algorithm

When designing algorithm	When writing algorithm
1) Output (What output u want)	1) Input
2) Input	2) Processing
3) Processing	3) Output

→ **Algorithm:- (Method / procedure)**
- N lang
- Psedocode
- flow chart
- program

Input:-
i) Sequence
ii) finite steps
iii) particular task (At least one output)

=> English like

=> Symbolic

→ Understanding => C++
Java => Compiler => M/L 0/1
C#

2) Conditional:-

```
a=5; b=10;      ] (2)
if (a<b)        ] (1)
c= a+b*2;       ] (4)
  display c ;   ]

display "msg";  ] (1)
```

[8 B.O]
→ Worst case

[4 B.O]
→ Best case

⇒

```
a=5 , b=10, c=50; ] (3)

if (a<b && c>40)  ] (3)

   cout<<a;
   cout<< b;      ] (05)
   c= a+b;
   cout<< c;

else
   cout<<"a+b"; ] (01)
```

[11 B.O]
→ worst case

[7 B.O]
→ Best case

⇒ $c < lg(n) < \sqrt{n} < n < nlogn < n^2 < n^3 < 2^n < n^n < n!$

Binary search

linear search

Merge sort
Quick sort
heap sort

Bubble
insertion
Selection

⇒ Fact (n)

For i=1 to n
| f = f*i
End For

return f;

⇒ Fact (n)

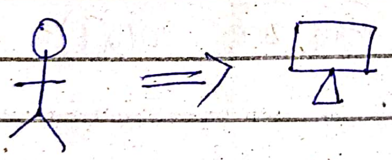
Input :- n : number

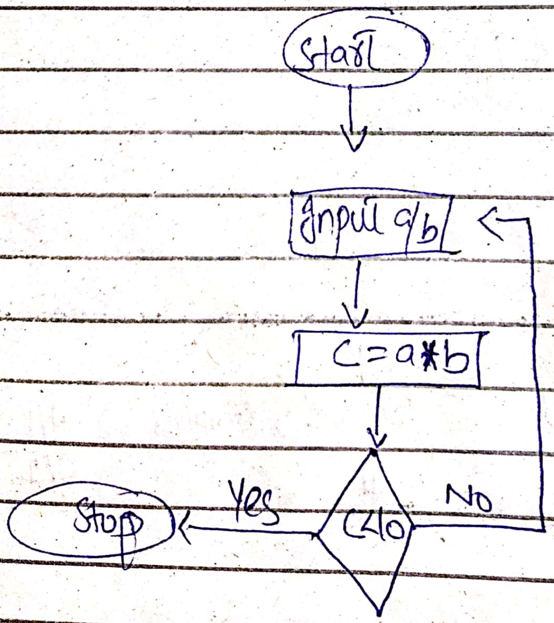
output :- f : factorial of n

⇒ 1) Input/get a b

2) Sum a,b // c=a+b

3) print sum





2/02/23 Wednesday

Emperical analysis:- $\xrightarrow{\text{Time + Cost}}$

HW	Same
NW	State of the system
PL	
PS	No of runs
OS	
tool	

Analytical analysis:-
↳ Mathematical approach
i) Identify basic operations
ii) frequency
iii) Sum up
iv) Map standard function

Algorithm/program:-
→ Sequential
→ Conditional
→ Repetition (loops)
→ functions
→ recursive functions.

i) Sequential:-

a=5, b=10;	2(1)	∴ 2+4+2+1+1
c = a*2 + b*3;	4(1)	= 10 B.O.
d = c/2;	2(1)	
cout << c;	1(1)	
cout << d;	1(1)	

Algo 2():-

```
{ a=5, b=10, c=50; ] (3)

  if (a<b) ] (1)

  if (a<c) ] (1)
      cout<<"a is better"; => (1)

  else
      if (b<c) ] (1)
          cout<< b; ] (1)
      else
          cout<< c; ] (1)
                                => (2)
}                                       => (3)
```

7 B.O

7 n

Algo(1):-

```
{
    algo 2 (); ] (7n)
    a=5, b=3; ] (2)
    if (a>b) ] (1)
        algo 2 (); ] 7n
}
```

=> 17 B.O

=> $\boxed{14n+3}$

23/02/23 Thursday

Example:-

∵ Algorithm => Max [A[1.....n]]

i) max = A[1]; 2(1) → (2)

ii) for (int i = 2; i <= n; i++)
1(1) 1(n-1) 2(n-1)

for i=2 to n ←

iii) if (max < A[i])
2(n-1)

Simply how many times

iv) max = A[i] ∵ $0 \leq K \leq (n-1)$
2(K)

O(n)

v) return max.
1(1)

∵ Direct method

$\Rightarrow 2+1+n-1+2(n-1)+2(n-1)+2K+1$

$\Rightarrow 4+5(n-1)+2K.$

∵ Best case (K=0) ∵ Worst case K=(n-1)

$4+5(n-1)+2(0)$ $4+5(n-1)+2(n-1)$

$4+5n-5$ $4+5n-5+2n-2$

$\boxed{5n-1}$ $\boxed{7n-3}$

↓ ↓

Best Worst

Alternate method:- condition, Increment, body.

$$\Rightarrow 2 + 1 + \sum_{i=2}^{n} (1 + 2 + 4) + 1$$

$$4 + \sum_{i=2}^{n} (\underbrace{7}_{\text{constant}})$$

$$4 + 7(n-2+1)$$

$$\boxed{7n-3}$$

$\because$ **formula:-** $\sum_{i=1}^{n} (\text{constant}) = \text{constant}(n-1+1)$

upper limit

lower limit

any value

2nd example:-

:- For loops

```
for (i=1; i<=n; i++)      1(1)   1(n)   2(n)
    print "Ali".          1(n)
```

$\Rightarrow$ Direct method:-

$$1+2n+n+n$$

$$\boxed{1+4n}$$

$\Rightarrow$ Alternate method:-

$$1 + \sum_{i=1}^{n} (1+2+1)$$

$$1 + \sum_{i=1}^{n} (4)$$

$$1 + 4(n-1+1)$$

$$\boxed{1+4n} \longrightarrow O(n)$$

Imp

Nested loops:-

Example:- $\overset{1(1)}{\text{for}(i=1;}\ \overset{1(n)}{i<=n;}\ \overset{2(n)}{i++)}$ → n

$\overset{1(1)}{\text{for}(j=i;}\ \overset{1(n)}{j<=n;}\ \overset{2(n)}{j++)}$

print "Ali" → n

1(n)

O(n^2)

Alternate method:-

$$1+\sum_{i=1}^{n}\left(1+2+1+\sum_{j=i}^{n}(1+2+1)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}(4)\right)$$

$$1+\sum_{i=1}^{n}\left(4+4(n-i+1)\right)$$

$$1+\sum_{i=1}^{n}(4+4n-4i+4)$$

$$1+\sum_{i=1}^{n}(8)+\sum_{i=1}^{n}(4n)-\sum_{i=1}^{n}(4i)$$

∵ Formula

$$1+8(n-\cancel{1}+\cancel{1})+4n(n-\cancel{1}+\cancel{1})-\overset{2}{\cancel{4}}\left(\frac{n(n+1)}{\cancel{2}}\right) \quad ∵ \sum_{i=1}^{n}(i)=\frac{n(n+1)}{2}$$

$$1+8n+4n^2-2n^2-2n \Rightarrow \boxed{1+6n+2n^2} \rightarrow O(n^2)$$

⇒ Assignment #01

"Home-work"

23/02/23 Thursday

Example:- for (int i=1; i<=n; i++)

for (j=1; j<i; j++)

print "Ali"

Solution:- $1+\sum_{i=1}^{n}\left(1+2+1+\sum_{j=1}^{i}(1+2+1)\right)$

$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{i}(4)\right)$

$1+\sum_{i=1}^{n}\left(4+4(i-\cancel{1}+\cancel{1})\right)$

$1+\sum_{i=1}^{n}(4+4i)$

$1+\sum_{i=1}^{n}(4)+\sum_{i=1}^{n}(4i)$

$1+4(n-\cancel{1}+\cancel{1})+\frac{\overset{2}{\cancel{4}}(n(n+1))}{\cancel{2}}$

$1+4n+2n^2+2n$

$\boxed{1+6n+2n^2} \rightarrow O(n^2)$

27/02/23 Monday

Nested loops:-

Example:- for(i=1; i<=n; i++)

for(j=1; j<=n; j++)

for(k=1; k<=n; k++)

Print "Ali".

Solution:- $1+\sum_{i=1}^{n}\left(1+2+1+\sum_{j=1}^{n}\left(1+2+1+\sum_{k=1}^{n}(1+2+1)\right)\right)$

$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{n}\left(4+\sum_{k=1}^{n}(4)\right)\right)$

$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{n}\left(4+4(n\cancel{-1}\cancel{+1})\right)\right)$

$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{n}(4+4n)\right)$

$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{n}(4)+\sum_{j=1}^{n}(4n)\right)$

$1+\sum_{i=1}^{n}\left(4+4(n\cancel{-1}\cancel{+1})+4n(n\cancel{-1}\cancel{+1})\right)$

$1+\sum_{i=1}^{n}\left(4+4n+4n^2\right)$

$1+4(n\cancel{-1}\cancel{+1})+4n(n\cancel{-1}\cancel{+1})+4n^2(n\cancel{-1}\cancel{+1})$

$\boxed{1+4n+4n^2+4n^3}$

Imp Formulas

i) $\sum_{i=1}^{n} (\text{constant}) = c(n-1+1)$

ii) $\sum_{i=1}^{n} (i) = \frac{n(n+1)}{2}$

iii) $\sum_{i=j}^{n} (i) = \sum_{i=1}^{n} i - \sum_{i=1}^{j} i$

iv) $\sum_{i=1}^{n} (i^2) = \frac{n(n+1)(2n+1)}{6}$

Example:- for(i=1; i<=n; i++)

for (j=i; j<=n; j++)

for(k=1; k<=n; k++)

print "Ali".

=> Solution:- $1 + \sum_{i=1}^{n} \left(1+2+1 + \sum_{j=i}^{n} \left(1+2+1 + \sum_{k=1}^{n} (1+2+1)\right)\right)$

$1 + \sum_{i=1}^{n} \left(4 + \sum_{j=i}^{n} \left(4 + \sum_{k=1}^{n} (4)\right)\right)$

$1 + \sum_{i=1}^{n} \left(4 + \sum_{j=i}^{n} \left(4 + 4(n - \cancel{1} + \cancel{1})\right)\right)$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}(4+4n)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}(4)+\sum_{j=i}^{n}(4n)\right)$$

$$1+\sum_{i=1}^{n}\left(4+4(n-i+1)+4n(n-i+1)\right)$$

$$1+\sum_{i=1}^{n}\left(4+4n-4i+4+4n^2-4ni+4n\right)$$

$$1+\sum_{i=1}^{n}\left(8+8n+4n^2-4i-4ni\right)$$

$$1+\sum_{i=1}^{n}(8)+\sum_{i=1}^{n}(8n)+\sum_{i=1}^{n}(4n^2)-\sum_{i=1}^{n}(4i)-\sum_{i=1}^{n}(4ni)$$

$$1+(8(n-\cancel{1}+\cancel{1}))+8n(n-\cancel{1}+\cancel{1})+4n^2(n-\cancel{1}+\cancel{1})-\frac{\overset{2}{\cancel{4}}(n(n+1))}{\cancel{2}}-\frac{\overset{2}{\cancel{4}}n(n(n+1))}{\cancel{2}}$$

$$1+8n+8n^2+4n^3-2n^2-2n-2n^3-2n^2$$

$$\boxed{1+6n+4n^2+2n^3}\rightarrow O(n^3)$$

=) Example:.

```
for( i=1;  i<=n; i++)
    for( j=1;  j<=i; j++)
        for( k=j;  k<=n; k++)
```

Solution:

$$1+\sum_{i=1}^{n}\left(1+2+1+\sum_{j=1}^{i}\left(1+2+1+\sum_{k=j}^{n}(1+2)\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{i}\left(4+\sum_{k=j}^{n}(3)\right)\right)$$

Scanned with CamScanner

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{i}\left(4+3(n-j+1)\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{i}\left(4+3n-3j+3\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=1}^{i}(7)+\sum_{j=1}^{i}(3n)-\sum_{j=1}^{i}(3j)\right)$$

$$1+\sum_{i=1}^{n}\left(4+7(i-\cancel{1}+\cancel{1})+3n(i-\cancel{1}+\cancel{1})-3\left(\frac{n(n+1)}{2}\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+7i+3ni-\left(\frac{3n^2+3n}{2}\right)\right)$$

$$1+\sum_{i=1}^{n}\left(\frac{8+14i+6ni-3n^2-3n}{2}\right)$$

$$1+\sum_{i=1}^{n}\left(\frac{8}{2}+\frac{14i}{2}+\frac{6ni}{2}-\frac{3n^2}{2}-\frac{3n}{2}\right)$$

$$1+\sum_{i=1}^{n}(4)+\sum_{i=1}^{n}(7i)+\sum_{i=1}^{n}(3ni)-\sum_{i=1}^{n}\left(\frac{3n^3}{2}\right)-\sum_{i=1}^{n}\left(\frac{3n}{2}\right)$$

$$1+4(n-\cancel{1}+\cancel{1})+7\left(\frac{n(n+1)}{2}\right)+3n\left(\frac{n(n+1)}{2}\right)-\frac{3n^2}{2}(n-\cancel{1}+\cancel{1})-\frac{3n}{2}(n-\cancel{1}+\cancel{1})$$

$$1+4n+\frac{7n^2}{2}+\frac{7n}{2}+\cancel{\frac{3n^3}{2}}+\cancel{\frac{3n^2}{2}}-\cancel{\frac{3n^3}{2}}-\cancel{\frac{3n^2}{2}}$$

$$1+4n+\frac{7n^2}{2}+\frac{7n}{2}$$

$$\boxed{1+\frac{15n}{2}+\frac{7n^2}{2}}\rightarrow O(n^2)$$

06/03/23 Monday

Square function

Example:

★ for(int i=1; i^2<=n; i++) → $O(\sqrt{n})$
print "Ali".

★ for(i=1 ; i<=n/2 ; i++) → $O(n/2)$
print"Ali".

★ for(i=n/2 ; i<=n ; i++) → $O(n/2)$
print"Ali".

★ for (i=1; i<=n ; i=i+2) → $O(n/2)$
print"Ali".

so,
i= 1, 3, 5, 7, ... n

★ for(i=1; i<= ; i=i*2) → $O(\log_2 n)$

so,
$i = 1, 2, 4, 8, \ldots n$
$i = 2^0, 2^1, 2^2, 2^3, \ldots 2^k$

$2^k = n$

$\log 2^k = \log n$

$k \log 2 = \log n$

$\boxed{k = \log_2(n)}$

Class-work Wednesday

01/03/23

Example:- for (i=1; i≤n; i++)

for (j=i; j≤n; j++)

for (k=1; k≤j; k++)

Print "Ali".

Solution:

$$1+\sum_{i=1}^{n}\left(1+2+1+\sum_{j=i}^{n}\left(1+2+1+\sum_{k=1}^{j}(1+2+1)\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}\left(4+\sum_{k=1}^{j}(4)\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}\left(4+4(j\cancel{-1}\cancel{+1})\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}(4+4j)\right)$$

$$1+\sum_{i=1}^{n}\left(4+\sum_{j=i}^{n}(4)+\sum_{j=i}^{n}(4j)\right)$$

$$1+\sum_{i=1}^{n}\left(4+4(n-i+1)+\overset{2}{\cancel{4}}\left(\frac{n(n+1)}{\cancel{2}}\right)\right)$$

$$1+\sum_{i=1}^{n}\left(4+4n-4i+4+2n^2+2n\right)$$

$$1+\sum_{i=1}^{n}\left(8+6n+2n^2-4i\right)$$

$$1+\sum_{i=1}^{n}(8)+\sum_{i=1}^{n}(6n)+\sum_{i=1}^{n}(2n^2)-\sum_{i=1}^{n}(4i)$$

$$1+8(n\cancel{-1}\cancel{+1})+6n(n\cancel{-1}\cancel{+1})+2n^2(n\cancel{-1}\cancel{+1})-\overset{2}{\cancel{4}}\left(\frac{n(n+1)}{\cancel{2}}\right)$$

* for(i=1; i<=n ; i++) →n

for(j=1; j<=n ; j=j*2) → log n.

for(k=1 ; k<=n; k=k*2) → log n

print("Ali").

∴ $\log n^2 \cdot n \Rightarrow O(n \log n^2)$.

Scanned with CamScanner

$1 + 8n + 6n^2 + 2n^3 - 2n^2 - 2n$

$1 + 8n - 2n^2 + 2n^3 + 6n^2 - 2n$ ∵ Rearragned

$\boxed{1 + 6n + 4n^2 + 2n^3} \rightarrow O(n^3)$

⇒ **lecture:** for(i=1; i^2<n; i++) ∵ $i^2 < n$

print "Ali". $i < \sqrt{n}$

⇓

So $O(\sqrt{n})$

* for (i=1; i<=n; i=i+2)

Print "Ali".

i= 1, 3, 5, ... n

* for (i=1; i<=n; i=i*2)

$$\begin{array}{cccccc} i= 1 & 2 & 4 & 8 & \ldots & n \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow \\ 2^0 & 2^1 & 2^2 & 2^3 & \ldots & 2^k \end{array}$$

$2^k = n$.

Taking log on b/s.

$\log 2^k = \log n$

$k \log 2 = \log n$

$\boxed{k = \log_2 n}$

* for(i=1; i<=n; i=i*3) → (O(log$_3$n))

print "Ali".

So, $i = 1, 3, 9, 27, \ldots, n$

$i = 3^0, 3^1, 3^2, 3^3, \ldots, 3^k$

$3^k = n$

$\log 3^k = \log n$

$k \log 3 = \log n$

$\boxed{k = \log_3 n}$

* for (i=1; i<=n; i++) → n

for(j=1; j<=n; j++) → n

for (k=1; k<=500; k++) → 500

$\therefore 500n^2 \rightarrow O(n^2)$

* for(i=1; i<=n/2; i++) → n/2

for(j=1; j<=n; j=j*2) → log n

print "Ali".

$\therefore \log n \cdot n/2 \Rightarrow O(n \log n)$

/03/2023 2nd class Monday

Algorithm => Complexity => $f(n)$ => $4n^2+5n+3$

Standard function(complexity class) => $\sqrt{n}, n^2, n\log n, 2^n$

Mapping Complexity ($f(n)$) on standard function.

Formula:- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C$

Conditions:-

i) $0 \le C < \infty \Rightarrow f(n) = \overset{\text{Big Ohh}}{O\, g(n)}$ => upper bounded

ii) $0 < C \le \infty \Rightarrow f(n) = \overset{\text{Big omega}}{\Omega\, g(n)}$ => lower bounded

iii) $0 < C < \infty \Rightarrow f(n) = \overset{\text{Theta.}}{\theta\, g(n)}$ => Tightly bounded

Main Course starts from here

Example:-

1) $f(n) = 4n^2+5n+3$
$g(n) = n$

By using formula:-

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \Rightarrow \lim_{n\to\infty} \frac{4n^2+5n+3}{n}$$

$$\Rightarrow \lim_{n\to\infty} \frac{4n^2}{n} + \frac{5n}{n} + \frac{3}{n}$$

$$\Rightarrow \lim_{n\to\infty} \left(4n + 5 + \frac{3}{n}\right)$$

Scanned with CamScanner

$$\Rightarrow \lim_{n\to\infty} 4n + \lim_{n\to\infty} 5 + \lim_{n\to\infty} \frac{3}{n}$$

$$\Rightarrow 4(\infty) + 5 + \frac{3}{\infty}$$

$$\Rightarrow \infty + 5 + 0 \boxed{\Rightarrow \infty}$$

So, $f(n) \neq O(g(n))$

$f(n) = \Omega(g(n))$

$f(n) \neq \Theta(g(n))$

2) $f(n) = 4n^2 + 5n + 3$

$g(n) = n^2$

$$\Rightarrow \lim_{n\to\infty} \frac{4n^2+5n+3}{n^2}$$

$$\Rightarrow \lim_{n\to\infty} 4 + \frac{5}{n} + \frac{3}{n^2}$$

$$\Rightarrow \cancel{l} \; 4 + 0 + 0 \boxed{\Rightarrow 4}$$

$f(n) = O(g(n))$

$4n^2 + 5n + 3 = O(g(n))$

$f(n) = \Omega(g(n))$

$f(n) = \Theta(g(n))$

3) $f(n) = 4n^2+5n+3$

$g(n) = n^3$

$\Rightarrow \lim_{n\to\infty} \frac{4n^2+5n+3}{n^3} \Rightarrow \lim_{n\to\infty} \frac{4n^2}{n^3} + \lim_{n\to\infty} \frac{5n}{n^3} + \lim_{n\to\infty} \frac{3}{n^3}$

$\Rightarrow 0+0+0$

$\boxed{\Rightarrow \quad 0}$

$f(n) = O(g(n))$

$f(n) \neq \Omega(g(n))$

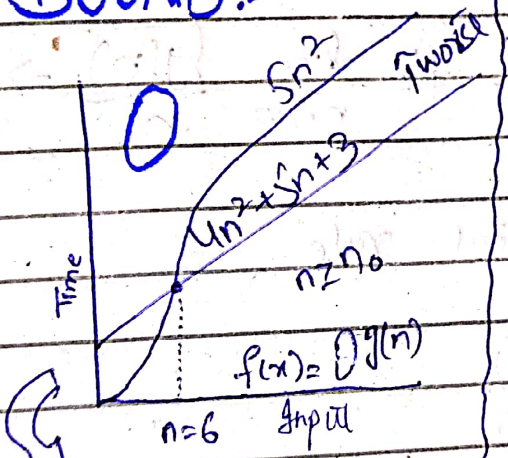
$f(n) \neq \Theta(g(n))$

⇒ UPPER BOUND:-

$f(n)=4n^2+5n+3$

$g(n)=5n^2$

for $n=6 \Rightarrow$ $f(n)$

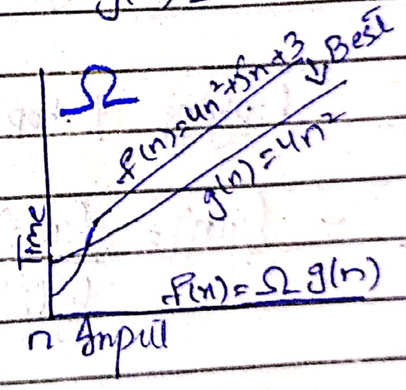


will not exceed standard function.

⇒ LOWER BOUND:-

$f(n)=4n^2+5n+3$

$g(n) = 4n^2$

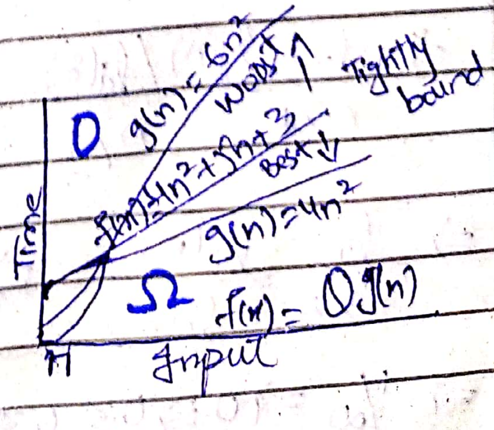


⇒ Tightly bound:-

$f(n)=4n^2+5n+3$

$g(n) = 4n^2$

$g(n) = 6n^2$



⇒ Behave same like

8/03/2023 Wednesday

Asymptotic notations:-

$O \Rightarrow$ Big oh

$\Omega \Rightarrow$ Big omega

$\Theta \Rightarrow$ Theta

* L-Hopital rule:-

Q:- Show that $\underbrace{lg(n)}_{f(n)} = O(\underbrace{n}_{g(n)})$

Solution:- $\lim_{n \to \infty} \frac{f(n)}{g(n)} \Rightarrow \lim_{n \to \infty} \frac{lg(n)}{n} \Rightarrow \frac{\infty}{\infty}$

Applying L.Hopital rule

$\because lg(n) = \frac{lg(n)}{lg.(2)}$

putting value of $lg(n)$

$\Rightarrow \lim_{n \to \infty} \frac{lg(n)/lg(2)}{n/1}$

$\Rightarrow \lim_{n \to \infty} \frac{lg(n)}{n.ln(2)} \Rightarrow \lim_{n \to \infty} \frac{1/n}{ln2.1} \Rightarrow \lim_{n \to \infty} \frac{1}{n.ln2}$

$\Rightarrow 1/\infty \Rightarrow 0 \Rightarrow 0 \leq 0 < \infty \Rightarrow$ proved $lg(n) = O(n)$

Q. Show that $n = O(2^n)$

Solution:- As we know the formula.

$$\Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)}$$

putting values.

$$\Rightarrow \lim_{n\to\infty} \frac{n}{2^n} \Rightarrow \frac{\infty}{\infty}$$

So the answer is not what we want so, applying L Hopital rule.

$$\Rightarrow \frac{1 \cdot dn/dn}{2^n \cdot \ln 2}$$

$$\Rightarrow \frac{1}{2^n \cdot \ln 2} \Rightarrow \frac{1}{2^\infty \cdot \ln 2} \Rightarrow \frac{1}{\infty} \Rightarrow 0$$

$$0 \leq 0 < \infty$$

Hence $n = O(2^n)$ proved

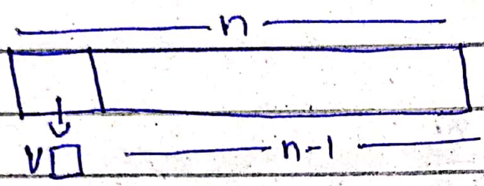
Scanned with CamScanner

15/03/2023 Wednesday

ALGORITHM DESIGN STRATEGIES:-

- Decrease and conquer
- Divide and conquer.

1) Decrease and conquer:-



=> Example:- if $(v == A[i])$ display msg.] => C] → how many time executed

↗ how much conditions

Recurrence relations:-

$T(n) = T(n-1) + C$

$T(0) = 0$

Telescopic sum:-

$T(n) = \cancel{T(n-1)} + c$

$\cancel{T(n-1)} = \cancel{T(n-2)} + c$

$\cancel{T(n-2)} = T(n-3) + c$

⋮

$\cancel{T(2)} = \cancel{T(1)} + c$

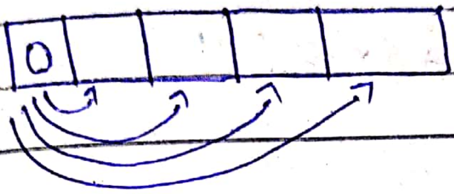
$\cancel{T(1)} = T(0) + c$

$T(n) = c + c + c + c$

$\boxed{T(n) = nc} \Rightarrow O(n)$

$$T(n) = T(n-1) + nc \quad \Big] \text{ n comparison}$$
$$T(0) = 0$$

for sorting an array.



$$T(n) = T(n-1) + nc$$
$$T(n-1) = T(n-2) + (n-1)c$$
$$T(n-2) = T(n-3) + (n-2)c$$
$$\vdots$$
$$T(2) = T(1) + 2c$$
$$T(1) = T(0) + 1c$$

$$T(n) = nc + (n-1)c + (n-2)c + \cdots + 2c + 1c$$

$$T(n) = c\,(n + (n-1) + (n-2) + \cdots + 1)$$

$$T(n) = c\left(\frac{n(n+1)}{2}\right)$$

$$\boxed{T(n) = O(n^2)}$$

$\Rightarrow$ $T(n) = T(n-1) + n^2 C$

$T(0) = 0$

$T(n) = T(n-1) + n^2 C$

$T(n-1) = T(n-2) + (n-1)^2 C$

$T(n-2) = T(n-3) + (n-2)^2 C$

$\vdots$

$T(2) = T(1) + (2)^2 C$

$T(1) = T(0) + (1)^2 C$

$T(n) = n^2 C + (n-1)^2 C + (n-2)^2 C + \dots 2^2 C + 1^2 C$

$T(n) = C\left(n^2 + (n-1)^2 + (n-2)^2 + \dots + (1)^2\right)$

$T(n) = C\left(\frac{n(n+1)(2n+1)}{6}\right)$

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

ii) DIVIDE & CONQUER:-

for binary search

3	5	7	8	9	5

< = >

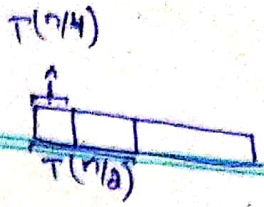
$\frac{1+6}{2} = \frac{7}{2} = 3$

```
BS (A, s, e, v)
{ while(s<=e) / if (s<=e)
      { mid= (s+e) /2
          { if (v== A[mid])
                 display msg
          else
               if (v< A[mid])
                   BS (A, S, mid-1, V)
               else BS( A, mid+1, e, v)
```

C

Input size

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



H·W

$$\boxed{\begin{array}{l} T(n) = 2T\left(\frac{n}{2}\right) + nc \\ T(1) = c \end{array}}$$

$$T(1) = c$$

$$\Rightarrow T(n) = T\left(\frac{n}{2}\right) + c \quad \text{—(i)} \qquad \Rightarrow T(n) = T\left(\frac{n}{2^1}\right) + 1c$$

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

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + c \quad \text{—(ii)}$$

put (ii) in (i)

$$T(n) = T\left(\frac{n}{4}\right) + 2c \quad \text{—(iii)} \qquad \Rightarrow T(n) = T\left(\frac{n}{2^2}\right) + 2c$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n/2}{4}\right) + c$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + c \quad \text{—(iv)}$$

put (iv) in (iii).

$$T(n) = T\left(\frac{n}{8}\right) + 3c \qquad \Rightarrow T(n) = T\left(\frac{n}{2^3}\right) + 3c$$

K-time division:-

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

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

$n = 2^k \Rightarrow k = \lg(n)$

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

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

$\boxed{T(n) = O(\lg(n))}$

20/03/23 Monday

Merge sort:-

K=S ....> e

B=

```
MS (A,S,e)
{
  if(S<e)
    mid= (S+e)/2 ;
    MS= (A, S, mid);
    MS= (A, mid+1, e);
    Merge (A, S, mid, e);
}

  Merge (A, S, mid, e);
   B[S ......e]
   int i;  K=j;  j=mid+1;
   while(i<=mid && j<=e)
     if(A[i] < A[j])
       B[K++]= A[i++];
     else
       B[K++] = A[j++];
     while (j<=mid)
       B[K++]= A[i++];
     while (j<=e)
       B[K++]= A[j++];
   for(int i=S; i<=e; i++)
       A[i]= B[i];
   }
```

Two sub-positions

Merge sort divided into two parts.

Original

$$T(n) = \underbrace{T\left(\tfrac{n}{2}\right) + T\left(\tfrac{n}{2}\right)}_{} + \overset{\text{cost of merging}}{cn}$$

each of size $n/2$.

$T(n) = 2T(n/2) + cn$ — (1)

$T(n) = 2^1 T(n/2^1) + 1cn$.

$T(1) = c$ → if there is only one value then some operations will perform/work that's why $T(1) = c$.

* Computing:-

$T(n/2) = 2T(n/4) + c(n/2)$ — (2).

put (2) in (1)

1) ⇒ $T(n) = 2\left[2T(n/4) + c(n/2)\right] + cn$

$T(n) = 4T(n/4) + 2cn$ — (3).

$T(n) = 2^2 T(n/2^2) + 2cn$

Now,

$T(n/4) = 2T(n/8) + c(n/4)$ — (4).

put (4) in (3)

3) ⇒ $T(n) = 4\left[2T(n/8) + c(n/4)\right] + 2cn$

$T(n) = 8T(n/8) + 3cn$ — (5)

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

Generaly :- $T(n) = 2^k T(n/2^k) + kcn$ — (A).

$\because \frac{n}{2^k} = 1$ $\because$ at least one time it will run.

$2^k = n$

$\log 2^k = \log n$

$k \log 2 = \log n$

$\boxed{k = \log n}$

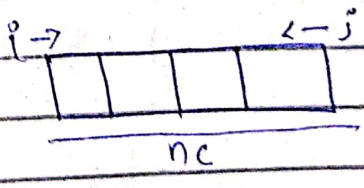
putting values in (A).

(A) $\Rightarrow T(n) = nT(1) + \log n(cn)$

$T(n) = nc + nc \log n$ $\because T(1) = c.$

$T(n) = O(n \log n)$

QUICK SORT:-



QS (A, s, e)
{ if (s < e)
{
p ← Partioning (A, s, e)
QS (A, s, p)
QS (A, p+1, e)
}
}

Partition:-

```
int partition (int* a, int low, int high)
{
    int pi = a[high];
    int j = low, i = low-1;
    while (j<=high-1)
    {
        if (a[j] <pi)
        {
            i++;
            std:: swap(a[i], a[j]);
        }
            j++;
    }
    std:: swap(a[i+1], a[high]);
    return i+1;
}
void quick sort (int* a, int low, int high)
{
    if (low <high)
    int pi = partition(a, low, high);
    quicksort (a, low, pi-1);
    quicksort( a, pi, high);
}
}
void display (int* a, int size)
{
    cout << "| ";
    for( int i=0; i<size; i++)
    { cout << a[i] <<" |";
    } cout<<"\n"; }.
```

Compiling:-

Method I:-

$T(n) = T(1) + T(n-1) + cn \quad -(1)$

n

2	5	7	9	11	12	15

$T(1) = C$

2		5	7	9	11	12	15

$T(1) + \quad T(n-1)$

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

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

$T(1) + \quad T(n-2)$

$\vdots$

$T(2) = T(1) + \cancel{T(1)} + 2C$

$T(1) = T(1) + T(0) + 1C$

$T(n) = [T(1) + T(1) + T(1) + \cdots + T(1)] + [cn + c(n-1) + c(n-2) + \cdots + 1c]$

$T(n) = [C + C + C + C + \cdots + C] + c[n + (n-1) + (n-2) + \cdots + 2 + 1]$

$T(n) = nC + c\left(\frac{n(n+1)}{2}\right)$

$\boxed{T(n) = O(n^2)}$

Method II:-

$T(n) = T(1) + T(n-1) + cn \quad -(1)$

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

put (2) in (1)

Scanned with CamScanner

1) $\Rightarrow T(n) = T(1) + T(1) + T(n-2) + c(n-1) + cn$ — (3)

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

put (4) in (3)

3) $\Rightarrow T(n) = T(1) + T(1) + T(1) + T(n-3) + c(n-2) + c(n-1) + c$

In generally:-

$T(n) = kc + T(n-k) + \cdots + c(n-2) + c(n-1) + cn$

$\because n-k = 1$

$\boxed{k = n}$

$\Rightarrow T(n) = kc + T(1) + \cdots + c(n-2) + c(n-1) + cn$

$T(n) = kc + c + \cdots + c(n-2) + c(n-1) + cn$

$T(n) = nc + c + \cdots + c((n-2) + (n-1) + \cdots + n)$

$T(n) = nc + c + \cdots + c\left(\frac{n(n+1)}{2}\right)$

$T(n) = nc + c\left(\frac{n(n+1)}{2}\right)$

$\boxed{T(n) = O(n^2)}$

Scanned with CamScanner

22/3/23 Wednesday

Master theorem:-

General:- $T(n) = aT\left(\frac{n}{b}\right) + cn^x$

if
$a < b^x \Rightarrow \Theta(n^x)$
$a = b^x \Rightarrow \Theta(n^x \log_b n)$
$a > b^x \Rightarrow \Theta(n^{\log_b a})$

Examples:-

i) $T(n) = 2T(n/2) + cn$

Sol:-
$a = 2$
$b = 2$
$x = 1$

$2 < 2^1$ ✗
$2 = 2^1$ ✓ $\Rightarrow \Theta(n \log_2 n)$
$2 > 2^1$ ✗

ii) $T(n) = 2T(n/2) + c$

Sol:-
$a = 2$
$b = 2$
$x = 0$

$2 < 2^0$ ✗
$2 = 2^0$ ✗
$2 > 2^0$ ✓ $\Rightarrow \Theta(n^{\log_2 2})$

iii) $T(n) = T(n/2) + c$

Sol: $a = 1$

$b = 2$

$x = 0$

$1 < 2^0$ ✗

$1 = 2^0$ ✓ $\Rightarrow \Theta(n^0 \log_2 n) \Rightarrow \Theta(\log_2 n)$

$1 > 2^0$ ✗

iv) $T(n) = 7T(n/2) + cn^2$

Sol: $a = 7$

$b = 2$

$x = 2$

$7 < 2^2$ ✗

$7 = 2^2$ ✗

$7 > 2^2$ ✓ $\Rightarrow \Theta(n^{\log_2 7})$

v) $T(n) = 2T(n/4) + cn^2$

Sol: $a = 2$

$b = 4$

$x = 2$

$2 < 4^2$ ✓ $\Rightarrow \Theta(n^2)$

$2 = 4^2$ ✗

$2 > 4^2$ ✗