Jasleen Kaur
1921049

Tutorial Sheet - 1

Q1 compute the Big Oh of the algorithm involving one for loop.

Ans) Time complexity of a loop is considered as $O(n)$ if the loop variables are incremented / decremented by a constant amount.

Example

	Cost	times
`int n=100`	C_1	1
`for (int i=0; i<n; i++)`	C_2	$n+1$
`{`		
`cout << "Hello";`	C_3	n
`}`		

Its Time Complexity will be

$$T_n = C_1 * 1 + C_2 * (n+1) + C_3 * (n)$$
$$= C_1 * 1 + C_2 n + C_2 + C_3 n$$
$$= (C_2 + C_3) n + (C_1 + C_2)$$

It is in the form of

$$T_n = an + b$$

So its Time complexity will be

$$\Rightarrow O(n)$$

Scanned with CamScanner

Page No:
Date:

Q2 Compute the time complexity of the following fragments of code

(a) for (i=n; i>1; i=i/2)
{
statement ;
}

Suppose i=n (given)
when i=8
$i = \frac{8}{2} = 4 \quad = \frac{n}{2^1}$

$\boxed{4 = \frac{n}{2}}$

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

$\frac{n}{2^3} \Rightarrow 1$

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

$n = 2^k$
Taking log on both sides

$\log_2 n = \cancel{\log_2} \log_2 2^k$
$\log_2 n = k \log_2$

$\boxed{\log_2 n = k}$ $\boxed{\log_2 = 1}$

So its time complexity is $O(\log_2 n)$

(b) for ($i=n$; $i>1$; $i=i*2$)
{
statement;
}

When $i=1$ $= 2^0$
$i = 1*2 = 2$ $= 2^1$
$i = 2*2 = 4$ $= 2^2$
$i = 4*2 = 8$ $= 2^3$
which implies
$\Rightarrow$ $2^k = n$

Taking $\log_2$ on both sides

$\log_2 2^k = \log_2 n$

$k \log_2 2 = \log_2 n$

$k \log_2 2 = \log_2 n$

$\boxed{\log_2 2 = 1}$

$\Rightarrow$ $k = \log_2 n$

~~to~~ So its time complexity is

$O(\log_2 n)$

Scanned with CamScanner