Isha Tirdal

PAGE

DATE

Tutorial - 1

Assignment

Q1. What do you understand by asymptotic notations? Define different asymptotic notation with example?

Asymptotic notations describes the running time of an algorithm. How much time an algo takes with the given input.

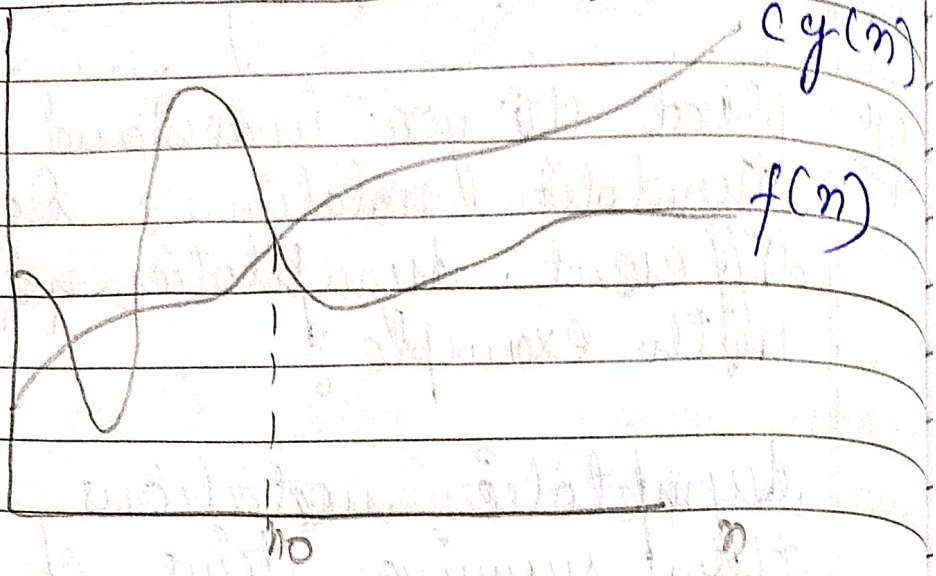
There are three different types of asymptotic notations

1. Big - O notation
2. Omega notation
3. Theta notation

Big - O notation (O - notation)
Big - O notation represent the upper bound of the running time of an algorithm. Thus,

PAGE
DATE

It gives the worst case of complexity of an algorithm.



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

if

$$f(n) \leq cg(n)$$
$$\forall\ n \geq n_0$$

Hence Big O gives the upper bound of a function

Omega notation (Ω- notation)

Omega notation represent the lower bound of the running time of an algorithm. Thus, it provides the best case complexity of an algorithm

PAGE
DATE

Q2:- what should be time complexity of

```
for(i=1 to n)
    { i = i * 2; }
```

$\sum_{i=1, i=i*2}^{n}$ $2 + 4 + 8 + 16 + 32 \cdots$

$\sum_{i=1}^{n}$ $2^1 + 2^2 + 2^3 + 2^4 + 2^5 \cdots 2^n$

$\sum_{i=1}^{n}$ 2^n

T.C = $O(2^n)$.

③ $T(n) = \begin{cases} 3T(n-1) & \text{if } n>0, \\ 1 & \text{otherwise} \end{cases}$

$T(n) = 3T(n-1)$ — ①

$T(0) = 1$

Let $n = n-1$

$T(n-1) = 3T(n-2)$ — ②

From eq^n ① & ②

PAGE
DATE

$T(n) = 3[3 \cdot T(n-2)]$ — (3)

Let $n = n-2$

$T(n-2) = 3T(n-3)$ — (4)

Put $T(n-2)$ in eq^n (3)

$T(n) = 3[3[3T(n-3)]]$

$T(n) = 3^3 \cdot T(n-3)$

$T(K) = 3^k T(n-k)$

$n - k = 0$

$n = 1 + k \Rightarrow k = n$

$T(n) = 3^n T(n-n)$

$T(n) = 3^n T(0)$

$T(n) = 3^n$

$T(n) = 3^n$

$T.C = O(3^n)$ Ans

PAGE _____
DATE _____

④ $T(n) = \begin{cases} 2T(n-1) - 1 \\ T(0) = 1 \end{cases}$

Sol^n :-

$T(n) = 2T(n-1) - 1$ — ①

Let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — ②

~~Let $n = n-2$~~ put $T(n-1)$ in eq^n ①

$T(n) = 2[2T(n-2) - 1] - 1$ — ③

Let $n = n-2$

from eq^n ①

$T(n-2) = 2T(n-3) - 1$ — ④

from eq^n ③ & ④

$T(n) = 2[2[2.[T(n-3) - 1] - 1] - 1$

$T(n) = 2^3 T(n-3) - 2^2 - 2^1 - 2$ — ⑤

Generalizing

$T(n) = 2^k T(n-k) - 2^{k-1} - 2^{k-2} - 2^{k-3} - \cdots - 2^0$

Let $n - k = 0$

$n = k$

PAGE

DATE

$$T(n) = 2^n T(n-n) - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^0$$

$$T(n) = 2^n T(0) - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^0$$

$$T(n) = 2^n \cdot 1 - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^0$$

$$T(n) = 2^n - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^0$$

$$T(n) = 2^n - [2^n - 1]$$

$$2^n - 2^n + 1$$

$$\boxed{T(n) = O(1)}$$

PAGE
DATE

Assignment

Q5. int i=1, s=1
while (s<=n)
{
i++;
s = s+i;
printf ("#");
}

i = 1	s = 1
i = 2	s = 3 = 1+2
i = 3	s = 6 = 1+2+3
i = 4	s = 10 = 1+2+3+4

$\sum_{s=1}^{?}$ s=s+i

1 + 3 + 6 + 10 ---- k

$$\frac{k(k+1)}{2} > n \quad [\because s <= n]$$

$$\frac{k(k+1)}{2} > n$$

$$\frac{k^2}{2} + \frac{k}{2} > n$$

lower limit just ignore

Constant → (2)

$$k^2 > n$$

PAGE _____
DATE _____

$k \geq \sqrt{n}$

T.C of the above is $O\sqrt{n}$
$T(n) = O(\sqrt{n})$

⑥ Time complexity of -

```
void function (int n)
{
  int i, count = 0;
  for(i = 1; i*i <= n; i++)
      count++;
}
```

Sol^n

$i = 1$	$1^2 = n$
$i = 2$	$4^2 = n$
$i = 3$	$9^2 = n$
$i = 4$	$16^2 = n$

$\sum_{i=1}^{n} 1 + 2^2 + 3^2 + 4^2 \cdots \; k^2 = n$

$i^2 i^2 i^2$

$\frac{k(k+1)}{2} \; 2 = n$

$k^2 \; 2 = n$

or $k\sqrt{2} = \sqrt{n}$

$T.C = O(\sqrt{n})$ Ans

PAGE
DATE

(7) Time complexity of

```
void function(int n)
{ int i, j, k, count = 0;
for(i = n/2; i <= n; i++)
{
for(j = 1; j <= n; j = j * 2)
for(k = 1; k <= n; k = k * 2)
    count++;
}
```

i	j	k
n/2	$\log n$	$(\log n)^2$
n/2+1	$\log n$	$(\log(n))^2$
⋮	⋮	⋮
upto n times	⋮	⋮
n	$\log n$	$[\log(n)]^2$

$\left(\frac{n}{2}+1\right)$ times

$$O(i*k) = O\left(\frac{n}{2}+1\right) * \left[\log_2 n\right]^2$$

$$T(n) = O\left(n(\log_2 n)^2\right)$$ Ans

PAGE _____
DATE _____

(8) Time complexity

```
function (int n)
{
  if (n == 1) return;
  for (i = 1 to n)
  {
    for (j = 1 to n)
    {
      printf ("**");
    }
  }
  3
  3
  3
  function (n-3);
}
```

Ans:- $T(n) = T(n-3) + n^2$ — (1)

let $n = n-3$

$T(n-3) = T(n-6) + (n-3)^2$ — (2)

from eqn (1) & (2)

$T(n) = T(n-6) + (n-3)^2 + n^2$ — (3)

let $n = n-6$

from eqn (1)

$T(n-6) = T(n-9) + (n-6)^2$ — (4)

from eqn (3) & (4)

$T(n) = T(n-9) + (n-6)^2 + (n-3)^2 + n^2$

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

PAGE
DATE

$(n-6)^2 + \cdots [n-(3(k-1)]^2$

Let $n-3k=1$

$n = 1+3k$

$k = (n-1)/3$

$T(n) = T(1) + n^2 + (n-3)^2 + (n-6)^2 \cdots [n-n+1)^2]$

$T(n) = 1 + n^2 + (n-3)^2 + (n-6)^2 \cdots 1$

$T(n) = cn^2 + k$ where c & k are constant.

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

(9) void function (int n)

{

for (i = 1 to n)

{

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

printf ("GG");

}

}

3 3

PAGE _____
DATE _____

$i = 1$, $j = 1, 2, 3, 4 \; \ldots \; n$ times

$i = 2$, $j = 1, 3, 5, 7 \; \ldots \; n/2$ times

$i = 3$ $j = 1, 4, 7, 11 \; \ldots \; n/3$ times

$i = n$ $j = 1 \; \ldots$ 1 times

i : n times

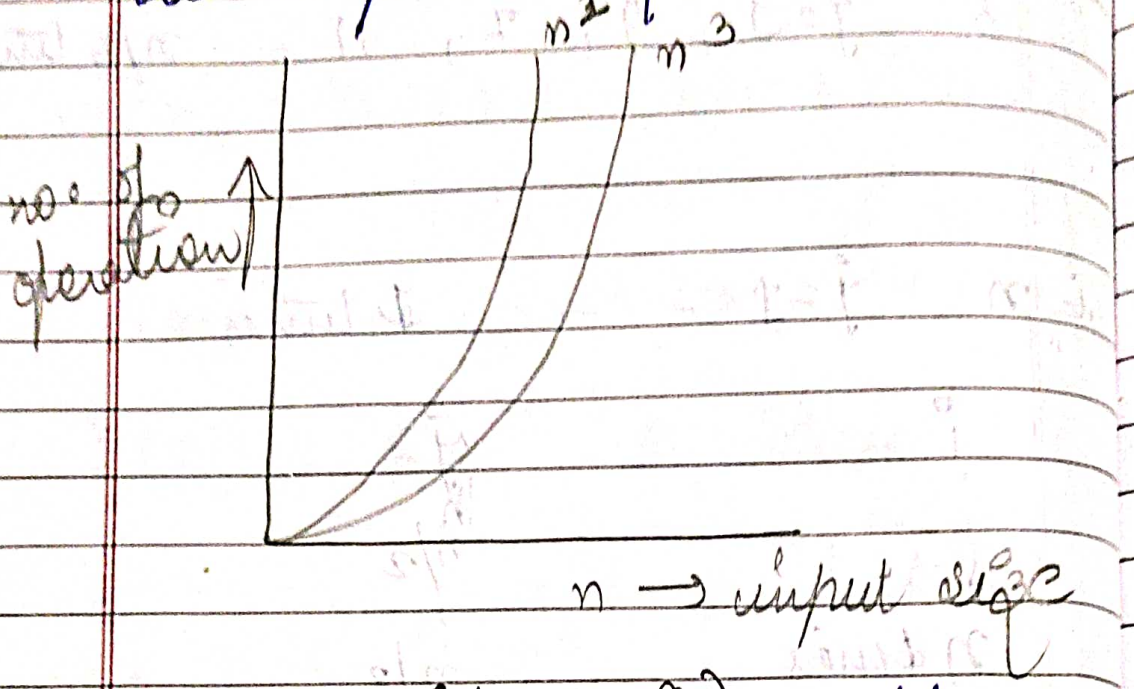
j : $n/2$, $n/3$, ... 1 times

$n(\log n)$

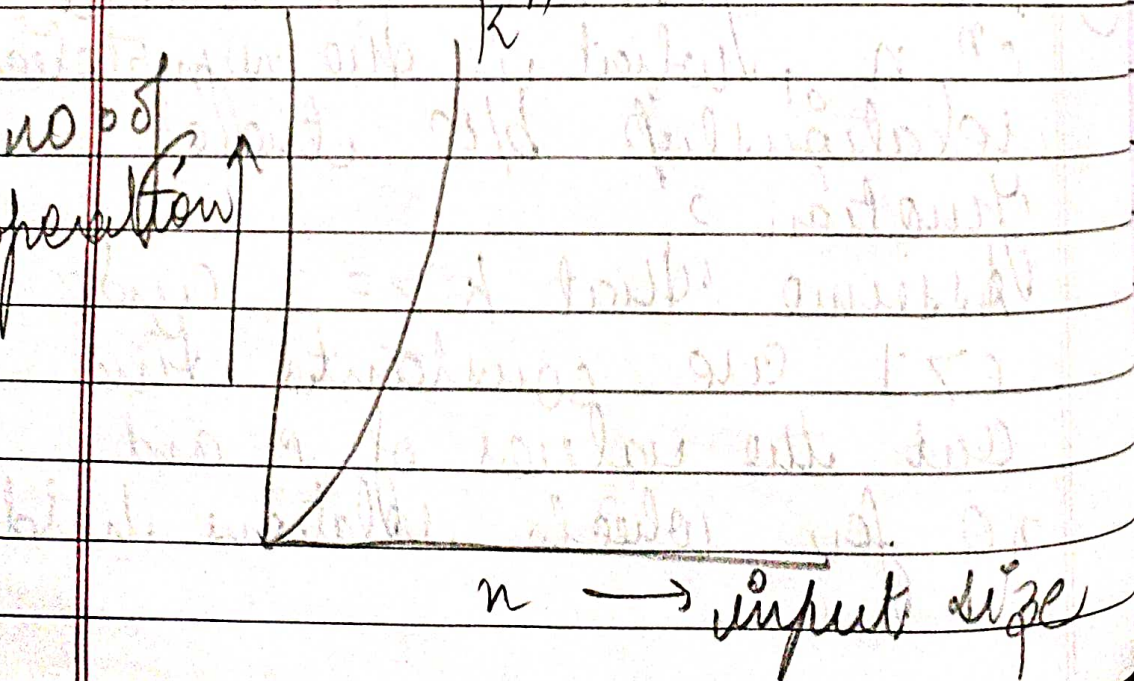
(10) For the function n^k and c^n n, what is the asymptotic relationship b/w these functions? Assume that $k >= 1$ and $c > 1$ are constants. Find out the values of c and n0 for which relation holds

PAGE
DATE

Def^n:- Polynomial (n^k):- When complexity grows proportion to the power of k of the input size



Exponential (k^n):- At each step the no. of operation becomes multiplied by k exponentially



PAGE _____
DATE _____

Relation b/w these two functions:-

n^k & c^n

Relation b/w n^k & c^n

$n^k = O(c^n)$

$n^k \leq a c^n$

$\forall\ n \geq n_0$ & some constant $a > 0$.

for

$n_0 = 1$

$c = 2$

$1^k \leq a\, 2^1$

$n_0 = 1$ & $c = 2$