CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC
Accredited with Grade A by KCG

CSE207 : Design and Analysis of Algorithn

Topic: Asymptotic Notations

GATE-CS-2001

Let $f(n) = n^2 \log n$ and $g(n) = n (\log n)^{10}$ be two positive fu
n. Which of the following statements is correct?

(A) f(n) = O(g(n)) and g(n) != O(f(n))

(B) f(n) != O(g(n)) and g(n) = O(f(n))

(C) f(n) = O(g(n)) but g(n) = O(f(n))

(D) f(n) != O(g(n)) but g(n) != O(f(n))

Answer: (B)

GATE CS 2008

Consider the following functions:

$f(n) = 2^n$

$g(n) = n!$

$h(n) = n^{\log n}$

Which of the following statements about the asymptotic of f(n), g(n), and h(n) is true?

(A) $f(n) = O(g(n))$; $g(n) = O(h(n))$

(B) $f(n) = \Omega(g(n))$; $h(n) = O(h(n))$

(C) $g(n) = O(f(n))$; $h(n) = O(f(n))$

(D) $h(n) = O(f(n))$; $g(n) = \Omega(f(n))$

GATE CS 2008 (continue)

(A) A

(B) B

(C) C

(D) D

Answer: (D)

GATE CS 2011

Which of the given options provides the increasing order asymptotic complexity of functions f1, f2, f3 and f4?

f1(n) = 2^n

f2(n) = n^(3/2)

f3(n) = nLogn

f4(n) = n^(Logn)

GATE CS 2011 (continue)

(A) f3, f2, f4, f1

(B) f3, f2, f1, f4

(C) f2, f3, f1, f4

(D) f2, f3, f4, f1

Answer: (A)

Gate IT 2008

Arrange the following functions in increasing asymptotic

A. $n^{1/3}$

B. e^n

C. $n^{7/4}$

D. $n \log^9 n$

E. 1.0000001^n

Gate IT 2008 (continue)

(A) A, D, C, E, B

(B) D, A, C, E, B

(C) A, C, D, E, B

(D) A, C, D, B, E

Answer: (A)

Self Assessment

Consider the following segment of C Code.

```
int j, n;
j=1;
While(j<=n)
    j=j*2;
```

The number of comparisons made in execution of the loo
n>0 is

Self Assessment (continue)

(A) $\lfloor log_2 n \rfloor * n$

(B) n

(C) $\lfloor log_2 n \rfloor + 1$

Answer : (c)

GATE-CS-2017

Consider the following functions from positives intege
numbers
$10, \sqrt{n}, n, \log_2 n, 100/n$.
The CORRECT arrangement of the above functions in i
order of asymptotic complexity is:

(A) $\log_2 n, 100/n, 10, \sqrt{n}, n$
(B) $100/n, 10, \log_2 n, \sqrt{n}, n$
(C) $10, 100/n, \sqrt{n}, \log_2 n, n$
(D) $100/n, \log_2 n, 10, \sqrt{n}, n$

Answer: (B)

Self Assessment

What is recurrence for worst case of QuickSort and what time complexity in Worst case?

(A) Recurrence is $T(n) = T(n-2) + O(n)$ and time complex $O(n^2)$

(B) Recurrence is $T(n) = T(n-1) + O(n)$ and time complex $O(n^2)$

(C) Recurrence is $T(n) = 2T(n/2) + O(n)$ and time compl $O(nLogn)$

(D) Recurrence is $T(n) = T(n/10) + T(9n/10) + O(n)$ and complexity is $O(nLogn)$

Answer: (B)

Self Assessment

What is the time complexity of fun()?

```
Int fun(int n)
{
    int count = 0;
    for (int i = 0;  i<n;  i++)
        for (int j = i;  j>0;  j--)
            count = count +1;
    return count;
}
```

Self Assessment (continue)

(A) Theta (n)

(B) Theta (n^2)

(C) Theta (n*Logn)

(D) Theta (nLognLogn)

Answer: (B)

Self Assessment

What is time complexity of fun()?

```
Int fun(int n)
{
    int count=0;
    for(int i=n;  i>0;  i/=2)
        for(int j=n;  j<i;  j++)
            count +=1;
    return count;
}
```

Self Assessment (continue)

(A) $O(n^2)$

(B) $O(n\log n)$

(C) $O(n)$

(D) $O(n\log n\log n)$

Answer: (B)

Self Assessment

What is the worst case time complexity of insertion sort position of the data to be inserted is calculated using bin search?

(A) N
(B) NlogN
(C) N^2
(D) N(logN)^2

Answer: (C)

GATE-CS-2003

Consider the following three claims

1. $(n+k)^m = \Theta(n^m)$, where k and m are constants
2. $2^{n+1} = O(2^n)$
3. $2^{2n+1} = O(2^n)$

Which of these claims are correct ?

(A) 1 and 2

(B) 1 and 3

(C) 2 and 3

(D) 1, 2, and 3

Answer: (A)

GATE-CS-2004

The recurrence equation evaluates to

$T(1) = 1$

$T(n) = 2T(n-1)+n, n>=2$

a. $2^{n+1} - n - 2$

b. $2^{n} - n$

c. $2^{n+1} - 2n - 2$

d. $2^{n} + n$

GATE-CS-2004 (continue)

(A) a

(B) b

(C) c

(D) d

Answer: (A)

Self Assessment

What is the time complexity of the below function?

```
void fun(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}
```

Self Assessment (continue)

(A) O(n)

(B) O(n^2)

(C) O(nlogn)

(D) O(n(logn)^2)

Answer: (A)

Self Assessment

In a competition, four different functions are observ
the functions use a single for loop and within the for
same set of statements are executed. Consider the fo
for loops:

a) for(i = 0; i < n; i++)

b) for(i = 0; i < n; i +=2)

c) for(i = 0; i < n; i *=2)

d) for(i = 0; i < -1; i /=2)

Self Assessment (continue)

If **n** is the size of input(positive), which function is most

the task to be performed is not an issue)?

(A) A

(B) B

(C) C

(D) D

Answer: (C)

Self Assessment

What does it mean when we say that an algorithm X is asymptotically more efficient than Y?

(A) X will be a better choice for all inputs

(B) X will be a better choice for all inputs except possibly
inputs

(C) X will be a better choice for all inputs except possibly
inputs

(D) Y will be a better choice for small inputs

Answer: (B)

Self Assessment

What does f(250,2) return?

```
F(m,n){
    ans = 1;
    count = 0;
    while (ans <=m)
    {
        count = count +1;
        ans = ans *n;
    }
    return(count)
}
```

Self Assessment (continue)

The loop executes till we reach a power of 2 that exceeds
the while condition, we have the invariant that ans == 2c
Since 28 is 256, the value of count is 8 when the function

Answer : 8

Self Assessment

Suppose $f(n)$ is $n^2 \log n$. Consider the following statemen

(A) $f(n)$ is $O(n \sqrt{n})$

(B) $f(n)$ is $O(n^2 \sqrt{n})$

(C) $f(n)$ is $O(n^3)$

Which of the following is true?

(i) (A), (B) and (C) are all not true.

(ii) (B) and (C) are true but (A) is not true.

(iii) (B) is true but (A) and (C) are not true.

(iv) (A) and (B) are true but (C) is not true.

Answer : (ii)

Self Assessment

An image processing application begins with two $n \times n$ ma
and B. The first phase of preprocessing the inputs takes (
steps for each of A and B. The second step involves a con
of A and B to yield a new matrix C in time $O(n^3)$. This is f
by an edge detection phase that takes times $O(n^2)$ for ma
What is the most accurate and concise description of the
complexity of the overall algorithm?

(a) $O(n^2)$

(b) $O(n^3)$

(c) $O(n^2+n^3)$

(d) $O(n^5)$

Self Assessment (Continue)

Answer: (b)

When there are multiple phases in sequence, the larg
phases dominates the overall complexity. Here the secon
$O(n^3)$ and the first and third phases are $O(n^2)$.

Self Assessment

If $T(n)$ is $O(n^2 \sqrt{n})$ which of the following is **false**?

(a) $T(n)$ is $O(n^2 \log n)$

(b) $T(n)$ is $O(n^3)$

(c) $T(n)$ is $O(n^3 \log n)$

(d) $T(n)$ is $O(n^4)$

Answer: (a)

Self Assessment

What does f(2000,3) return??

```
F(m,n) {
    ans = 1;
    count =0;
    while (ans <=m){
        count =count +1;
        ans =ans *n;
    }
    return(count)
}
```

Self Assessment (continue)

Answer

The loop executes till we reach a power of 3 that exceeds the while condition, we have the invariant that ans == 3^{c} 3^6 is 729 and 3^7 is 2187, the value of count is 7 when the returns.

Self Assessment

Suppose f(n) is $252n^3+164n^2+507$ and g(n) is $n^4 + 5n + 1$
h(n) be a third, unknown function. Which of the followin
is **not** possible.

(a) h(n) is O(f(n)) and h(n) is also O(g(n))

(b) h(n) is O(f(n)) but h(n) is not O(g(n))

(c) h(n) is O(g(n)) but h(n) is not O(f(n))

(d) h(n) is not O(f(n)) and h(n) is also not O(g(n))

Answer: (b)

Since f(n) is O(g(n)), if h(n) is O(f(n)) it must also be O(g(n

Self Assessment

Suppose our aim is to sort an array in ascending order. W
the following statements is true?

(a) Input in descending order is worst case for both selec
and insertion sort.

(b) Input in descending order is worst case for selection s
not for insertion sort.

(c) Input in ascending order is worst case for both selecti
and insertion sort.

(d) Input in ascending order is worst case for insertion so
for selection sort.

Answer: (a)

Self Assessment

Master's Theorem is used for

(a) Solving recurrences

(b) Solving iterative solutions

(c) Analyzing loops

(d) Calculating time complexity of any code

Answer: (a)

Self Assessment

Suppose we are sorting an array of eight integers using q
and we have just finished the first partitioning with the a
looking like this: 62 15 21 77 79 112 61 80

Statement 1: 77 could be a pivot.

Statement 2: 112 could be a pivot.

(a) Statement 1 and Statement 2 both are correct.

(b) Only Statement 1 is correct.

(c) Only Statement 2 is correct.

(d) Statement 1 and Statement 2 both are incorrect

Answer: (d)

Self Assessment

In quick sort of the following numbers, if the pivot is cho
first element, what will be the order of the numbers afte
partition function? Assume we are sorting in increasing
18, 8, 10, 21, 7, 2, 32, 6, 19).

(A) 8, 10, 7, 2, 6, 13, 18, 21, 32, 19

(B) 2, 6, 7, 8, 10, 13, 18, 19, 21, 32

(C) 6, 8, 10, 7, 2, 13, 18, 21, 32, 19

(D) 6, 8, 10, 7, 2, 13, 21, 32, 18, 19

Answer: (d)

Self Assessment

Choose the most appropriate option.

Which of the following option is correct?

Statement 1: The worst case of quick sort is when the inp
is in sorted order.

Statement 2: The worst case of quick sort is when the inp
is in reverse sorted order.

Self Assessment (continue)

(A) Only Statement 1 is correct

(B) Only Statement 2 is correct

(C) Statement 1 and Statement 2 both are correct

(D) Statement 1 and Statement 2 both are incorrect

Answer: (c)

Self Assessment

For the best case input, the running time of an insertion
algorithm is………

(A) Linear

(B) Binary

(C) Quadratic

(D) Depends on the input

Answer: (a)

Self Assessment

Which of the following examples represent the worst cas
for an insertion sort?

(A) Array in sorted order

(B) Array sorted in reverse order

(C) Normal unsorted array

(D) Large array

Answer: (b)

Self Assessment

Let $W(n)$ and $A(n)$ denote respectively, the worst case a average case running time of an algorithm executed on a size n, Which of the following is ALWAYS TRUE?

(A) $A(n) = \Omega(W(n))$

(B) $A(n) = \Theta(W(n))$

(C) $A(n) = O(W(n))$

(D) $A(n)$ and $W(n)$ are not comparable in terms of asymp notations.

Ans: C

Self Assessment

Consider the following two functions. What are time com
of the functions?

```
int fun1(int n)
{
   if (n <= 1) return n;
   return fun1(n-1);
}
```

```
int fun2(int n)

{

   if (n <= 1) return n;

   return fun2(n-1) + fun2(n
}
```

Self Assessment (continue)

(A) O(2^n) for both fun1() and fun2()

(B) O(n) for fun1() and O(2^n) for fun2()

(C) O(2^n) for fun1() and O(n) for fun2()

(D) O(n) for both fun1() and fun2()

Answer: (B)

Self Assessment

Consider the following segment of C-code:

```
int j, n;
 j=1;
While (j <= n)
    j=j*2;
```

Self Assessment (continue)

The number of comparisons made in the execution of the

any n > 0 is:

Base of Log is 2 in all options.

(A) CEIL(logn) + 2

(B) n

(C) CEIL(logn)

(D) FLOOR(logn) + 2

Answer: (D)

Self Assessment (continue)

For example, for n=5 we have the following (4) comparis

1 <= 5 (T)

2 <= 5 (T)

4 <= 5 (T)

8 <= 5 (F)

FLOOR(log_2 n)+2 = FLOOR(log_2 5) + 2 = FLOOR(2.3) +

= 4

Self Assessment

Consider the following C-program fragment in which
are integer variables.

```
for (i = n, j = 0; i >0; i /= 2, j += i);
```

Let val(j) denote the value stored in the variabl
termination of the for loop. Which one of the following is

(A) val(j) = Θ(logn)

(B) val(j) = Θ(sqrt(n))

(C) val(j) = Θ(n)

(D) val(j) = Θ(nlogn)

Self Assessment (Continue)

Answer (C)

Note the semicolon after the for loop, so there is nothing
body. The variable j is initially 0 and value of j is sum of v
i is initialized as n and is reduced to half in each iteration

$$j = n/2 + n/4 + n/8 + .. + 1 = \Theta(n)$$

GATE-CS-2014

The minimum number of comparisons required to find t
minimum and the maximum of 100 numbers is __________

(A) 148
(B) 147
(C) 146
(D) 140

Answer: (A)

GATE-CS-2014 (Continue)

Steps to find minimum and maximum element out of n n

1. Pick 2 elements(a, b), compare them. (say a > b)
2. Update min by comparing (min, b)
3. Update max by comparing (max, a)

Therefore, we need 3 comparisons for each 2 elements, s
number of required comparisons will be (3n)/2 – 2, beca
do not need to update min or max in the very first step.

By putting the value n=100, (3*100/2)-2 = **148** which is

GATE-CS-2005

The space complexity of the above function is:

```
double foo (int n)
{
    int i;
    double sum;
    if (n = = 0) return 1.0;
    else{
        sum = 0.0;
        for (i = 0; i < n; i++)
            sum += foo (i);
        return sum;
    }
}
```

(A) O(1)

(B) O(n)

(C) O(n!)

(D) O(n^n)

Answer: (B)

Explanation: Note that the function foo() is recursive. Sp
complexity is O(n) as there can be at most O(n) active fu
(function call frames) at a time.

GATE-CS-2004

Two matrices M1 and M2 are to be stored in arrays A anc
respectively. Each array can be stored either in row-majc
column-major order in contiguous memory locations. Th
complexity of an algorithm to compute M1 × M2 will be

(A) best if A is in row-major, and B is in column- major o

(B) best if both are in row-major order

(C) best if both are in column-major order

(D) independent of the storage scheme

Answer: (D)

Self Assessment

For each group of functions, sort the functions in increas
of asymptotic (big-O) complexity:

$f_1(n) = n^{0.999999}\log n$
$f_2(n) = 10000000n$
$f_3(n) = 1.000001^n$
$f_4(n) = n^2$

Self Assessment (Continue)

Solution: The correct order of these functions is $f_1(n)$, f_2
$f_3(n)$. To see why $f_1(n)$ grows asymptotically slower then
recall that for any $c>0$, $\log n$ is $O(N^c)$. Therefors we have:

$f_1(n) = n^{0.999999}\log n = O(n^{0.999999} * n^{0.000001}) = O(n) = O(f_2($

The function $f_2(n)$ is linear, while the function $f_4(n)$ is qu
so $f_2(n)$ is $O(f_4(n))$.
Finally, we know that $f_3(n)$ is exponential, which grows
faster than quadratic, so $f_4(n)$ is $O(f_3(n))$.

Self Assessment

For each group of functions, sort the functions in increas

of asymptotic (big-O) complexity:

$$
\begin{aligned}
f_1(n) &= 2^{2^{100000}} \\
f_2(n) &= 2^{100000n} \\
f_3(n) &= \binom{n}{2} \\
f_4(n) &= n\sqrt{n}
\end{aligned}
$$

Self Assessment (Continue)

Solution:

The correct order of these functions is $f_1(n)$, $f_4(n)$, $f_3(n)$, f
variable n never appears in the formula for $f_1(n)$, so desp
multiple exponentials, $f_1(n)$ is constant. Hence, it is asym
smaller than $f_4(n)$, which does grow with n. We may rew
formula for $f_4(n) = n\sqrt{n} = n^{1.5}$.

The value of $f_3(n) = \binom{n}{2}$ is given by the formula n(n-1)/2
$\Theta(n^2)$. Hence, $f_4(n) = n^{1.5} = O(n^2) = O(f_3(n))$. Finally, $f_2(n)$
exponential, while $f_3(n)$ is quadratic, meaning that $f_3(n)$ i

Gate IT 2005

Let $T(n)$ be a function defined by the recurrence

$T(n) = 2T(n/2) + \sqrt{n}$ for $n \geq 2$ and

$T(1) = 1$

Which of the following statements is TRUE?

(A) $T(n) = \theta(\log n)$

(B) $T(n) = \theta(\sqrt{n})$

(C) $T(n) = \theta(n)$

(D) $T(n) = \theta(n \log n)$

Answer: (C)

GATE-CS-2016

Assume that the algorithms considered here sort the inp
sequences in ascending order. If the input is already in as
order, which of the following are TRUE ?

I. Quicksort runs in $\Theta(n2)$ time

II. Bubblesort runs in $\Theta(n2)$ time

III. Mergesort runs in $\Theta(n)$ time

IV. Insertion sort runs in $\Theta(n)$ time

GATE-CS-2016 (Continue)

(A) I and II only

(B) I and III only

(C) II and IV only

(D) I and IV only

Answer: (D)

Self Assessment

The auxiliary space of insertion sort is O(1), what does O
?

(A) The memory (space) required to process the data is
constant.

(B) It means the amount of extra memory Insertion Sort
consumes doesn't depend on the input. The algorithm sh
the same amount of memory for all inputs.

(C) It takes only 1 kb of memory .

(D) It is the speed at which the elements are traversed.

Answer: (B)

GATE CS 1999

If T1 = O(1), give the correct matching for the following p

(M) $T_n = T_n - 1 + n$	(U) $T_n = O(n)$
(N) $T_n = T_n /2 + n$	(V) $T_n = O(n\log n)$
(O) $T_n = T_n /2 + n\log n$	(W) $T = O(n^2)$
(P) $T_n = T_n - 1 + \log n$	(X) $T_n = O(\log n^{2n})$

(A) M-W N-V O-U P-X

(B) M-W N-U O-X P-V

(C) M-V N-W O-X P-U

(D) M-W N-U O-V P-X

Answer: (B)

GATE CS 1999 (Continue)

(M) T(n) = T(n-1) + n = 1 + 2 + 3 + … + n = O(n^2) — cho

(N) T(n) = T(n/2) + n = O(n), using master theorem case
choice is (U)

(O) T(n) = T(n/2) + nlogn = O(nlog^2 n), using master th
case -2, - choice is (X)

(P) T(n) = T(n-1) + logn = log 1 + log 2 + log 3 + … + log
= log(1*2*3…*n) = log(n!) = nlogn = O(nlogn) - cl
(v)

GATE CS 1998

Give the correct matching for the following pairs:

A.	$O(\log n)$	1. Selection of element in liner search
B.	$O(n)$	2. Insertion sort
C.	$O(n\log n)$	3. Binary search
D.	$O(n^2)$	4. Merge sort

Codes:

	A	B	C	D
a.	3	1	2	4
b.	3	1	4	2
c.	1	3	4	2
d.	1	4	3	2

GATE CS 1998 (Continue)

(A) a

(B) b

(C) c

(D) d

Answer: (B)

GATE CS 1996

Quicksort is run on two inputs shown below to sort in as
order taking first element as pivot,

(i) 1, 2, 3,......., n

(ii) n, n-1, n-2,......, 2, 1

Let C_1 and C_2 be the number of comparisons made for th
(i) and (ii) respectively. Then,

GATE CS 1996 (continue)

(A) $C1 < C2$

(B) $C1 > C2$

(C) $C1 = C2$

(D) We cannot say anything for arbitrary n

Answer: (C)

GATE CS Mock 2018

What is the time complexity of following function fun()? Assume that log(x) returns log value in base 2.

```
void fun()
{
   int i, j;
   for (i=1; i<=n; i++)
      for (j=1; j<=log(i); j++)
         printf("GeeksforGeeks");
}
```

(A) Θ(n)

(B) Θ(nLogn)

(C) Θ(n^2)

(D) Θ(n^2(Logn))

Answer: (B)

Explanation: The time complexity of above function can
written as: $\Theta(\log 1) + \Theta(\log 2) + \Theta(\log 3) + \ldots + \Theta(\log$ n
Θ (log n!)
Order of growth of 'log n!' and 'n log n' is same for large v
n, i.e., Θ (log n!) = Θ(n log n). So time complexity of fun()
n).

ISRO CS 2017

Consider the recurrence equation

$T(n) = 2T(n-1)$, if $n>0$

$=1$, otherwise

Then $T(n)$ is (in big O order)

(A) $O(n)$

(B) $O(2^n)$

(C) $O(1)$

(D) $O(\log n)$

Answer: (B)

ISRO CS 2017 (Continue)

Explanation: Using substitution Method:

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

$= 2(2T(n-2))= 2^2T(n-2)$

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

$=2(2^{n-3}T(n-(n-2)))=2^{n-3}T(n-(n-2))$

$= 2(2^{n-2}T(n-(n-1)))=2^{n-1}T(n-(n-1))=2^{n-1}T(1)$

$= 2(2^{n-1}T(n-(n)))=2^nT(n-(n))=T(0)$

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

So, option (B) is correct.

UGC-NET CS 2017 Nov – III

Consider the recurrence relation:

$T(n) = 8T(n/2) + C_n$, if $n>1$

$= b$, if $n=1$

Where b and c are constants.

The order of the algorithm corresponding to above recurrenc

is:

(A) n

(B) n^2

(C) $n \log n$

(D) n^3

Answer: (D)

UGC-NET CS 2017 Nov – III (Continue)

$T(n) = aT(n/2) + \Theta(nklogpn)$

In given question:

$T(n) = 8T(n/2) + Cn$

here $a = 8$ and $b = 2$ and $k = 1$.

clearly $a > bk$

So $T(n) = \Theta(nlogba\ )$

$T(n) = \Theta(nlog2\ 8)$

ie $T(n) = \Theta(n3)$

UGC NET CS 2017 Jan – III

The asymptotic upper bound solution of the recurrence r
given by
$T(n) = 2T(n/2) + n/\log n$ is:

(A) $O(n^2)$
(B) $O(n \lg n)$
(C) $O(n \lg \lg n)$
(D) $O(\lg \lg n)$

Answer: (C)

UGC NET CS 2017 Jan – III (Continue)

We will apply master theorem in this question:

$a = 2$, $b = 2$ and $k = 1$ and $p = -1$.

we will compare a and b^k

ie.

$2 = 2^1$ and we have $p = -1$

Then $T(n) = (n^{\log 2} \log \log n)$

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

And $(n \log \log n) >= O(n \log \log n)$

So, option (C) is correct.

ISRO CS 2015

The time complexity of the following C function is (assun

```
int recursive(int n){
if(n==1)
return(1) ;
Else
Return (recursive(n-1)+ recursive(n-1));
}
```

ISRO CS 2015 (Continue)

(A) $O(n)$

(B) $O(n \log n)$

(C) $O(n^2)$

(D) $O(2^n)$

Answer: (D)

ISRO CS 2015 (Continue)

Recurrence relation for the code = $T(n) = 2T(n-1) + k$.

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

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

$= 2(2(2T(n-3)+k)+k)+ k\ldots\ldots$

$= 2^{x}T(n-x) + 2(1 + 2 + \ldots.+ 2^{x-1})k$

When base condition is met, i.e. $n=1$, $x=n-1$

$= 2^{n-1} T(1)+ k()$

$=O(2^{n})$