DAA

- Algorithm :- An algorithm is well-defined computational procedure that takes some value, or set of values as input and produces some value, or set of values as output.

- characteristics of an algorithm
1. finiteness
2. Definiteness
3. Input
4. Output
5. effectiveness

☆ Performance analysis of algorithm

- Linear search :- It is a method for finding a particular value from the given list.
- It is a special case of brute-force search

Algo:- array A, element

```
for i = 1 to last index of A :
        if A[i] equals element:
              return i
return -1
```

Page No.
Date :

Bounds on running time

- **Lower bound :-** Lower bound of an algorithm defines the minimum time required, it is not possible to have any other algorithm (for the same problem) whose time complexity is less than $L(n)$ for random input.

- **Upper bound :-** An upper bound $U(n)$ of an algorithm defines the maximum time required, we can solve the problem in at most $U(n)$ time. Time taken by a algorithm to solve a problem with worst case input gives the upper bound.

★ Asymptotic notations

- It allow us to analyze an algorithm's running time.
- This is also known as an algorithm's growth rate.

1. O - Notation (Big O) (Upper bound)

$f(n) \leq c\, g(n)$ for all $n_0 \leq n$

- $g(n)$ is an asymptotically upper bound for $f(n)$.
- $f(n) = O(g(n))$

2. Ω - Notation (omega) (Lower bound)

$f(n) \geq c \cdot g(n)$

3. Θ - Notation (same order)

$c_1 g(n) \leq f(n) \leq c_2 g(n)$

Page No.
Date :

1. int_count (int n)
{
    sum = n(n+1)/2   1
    return sum   1
}

- step count $T(n) = 2$

2. int_count (int n)
{
    sum = 0   1
    for i = 1 to n {   n + 1
      sum += n + 1 - i }   n
    return sum   1
}

- Total step count $= 1 + n + 1 + n + 1$
$= 2n + 3$

3. int_count (int n)
{ sum = 0   1
    for (i = 1 to n) {   n + 1
    for (j = 1 to m) {   m(n+1)
      sum ++ }   nm
    }
    return sum   1
}

- Total stepcount $= 1 + n + 1 + nm + m + nm + 1$
$= 2nm + n + m + 3$

$T(n) = O(nm)$

Page No.
Date :

4. Largest number

Let max = x_i	1
for (i = 1 to n) {	n+1
do if (x_i > max)	n
max = x_i	n
print (max)	1

• Total step count = 1 + n + 1 + n + n + 1
= 3n + 3
T(n) = O(n)

5. Fun ()

	cost	#
{ sum = 0	C_1	1
for i = 0 to n−1	C_2	n+1
sum += a[i]	C_3	n
print(sum)	C_4	1
}		

• Total step count = $C_1 + nC_2 + C_2 + nC_3 + C_4$
$= (C_2 + C_3)n + (C_1 + C_2 + C_4)$
$= an + b$
T(n) = O n

6. factorial (n)

	Cost	#
{		
f = 1	C_1	1
for (i = 2 to n)	C_2	n
{ f = f * i	C_3	n − 1
return f	C_4	1
}		

Page No.
Date :

- Total step count = $c_1 + nc_2 + c_3 n + c_3 + c_4$

$= n(c_2 + c_3) + (c_1 - c_3 + c_4)$

$= an + b$

$T(n) = O(n)$

7. fibo(n)

```
{  let a[0] = 0
       a[1] = 1
   for i = 2 to n-1
   a[2] = a[i-1] + a[i-2]
   print[a[i]]
}
```

let a[0] = 0	c_1	1
a[1] = 1	c_2	1
for i = 2 to n-1	c_3	$n-1$
a[2] = a[i-1] + a[i-2]	c_4	$n-2$
print[a[i]]	c_5	1

- Total step count = $c_1 + c_2 + nc_3 - c_3 + nc_4 - 2c_4 + c_5$

$= n(c_3 + c_4) + (c_1 + c_2 - c_3 - 2c_4 + c_5)$

$= an + b$

$T(n) = O(n)$

8. fun(n)

```
{  for(i=1; i<=n; i++)
   { for(j=1; j<=n; j++)
   {  a = b+1 - j
   }
}
```

for(i=1; i<=n; i++)	c_1	$n+1$
for(j=1; j<=n; j++)	c_2	$n(n+1)$
a = b+1 - j	c_3	$n(n)$

- Total statement = $c_1(n) + c_1 + n^2 c_2 + nc_2 + n^2 c_3$

$= (c_2 + c_3)n^2 + n(c_1 + c_2) + c_1$

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

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

Page No.
Date :

256
16
4
2

11. Func()
{ while (n > 1)
  { n = $\sqrt{n}$ }
}

$n = 64$
$n = 8$
$n = 2\sqrt{2}$

$n = n^{1/2}$
$n = n^{1/4}$
$n = n^{1/8}$
$n = n^{1/2^k}$

$n^{1/2^k} = 2$

$\log_2 n^{1/2^k} = \log_2 2$

$\frac{1}{2^k} \log_2 n = 1$

$\log n = 2^k$

$\log_2 (\log_2 n) = k \log 2$

$\log_2 (\log_2 n) = k^2$

$T(n) = O(\log_2 (\log_2 n))$

12. func()
{ while (n > 2)
  { n = $n^{1/20}$ }
}

Page No.
Date :

$n = 5$

13.
```
fun()
{ while(i<n)
{   i = 2*i }
}
```

i	
1	$i = 2$
2	4
3	8
4	16
5	32

$2 \cdot n$

~~n~~ + i

~~nlog n~~

14.
```
fun()
{ for(i=1; i<n; i++)
  { for (j=1; j<n; j=2*j)
  { x=y+z }
  }
}
```

$n+1$

$n(\log_2 n)$

1

- $n \log_2 n$

Extra

1. for(i=0; i<n; i++) $\quad n+1$

{ sta... $\quad n$

}

– $O(n)$

2. for(i=0; i<n; i++) $\quad n+1$

{

for(j=0; j<n; j++) $\quad n(n+1)$

{ stat; $\quad n^2$

}

}

– $O(n^2)$

3. for(i=0; i<n; i++)

{ for(j=0; j<i; j++)

{ ; }

}

– $O(n^2)$

4. p=0

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

{ p = p+i;

}

i	p
1	0+1=1
2	1+2=3
3	1+2+3
⋮	⋮
k	

$\Rightarrow p = \frac{k(k+1)}{2}$

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

$k^2 > n$

$k > \sqrt{n}$

$\boxed{O\sqrt{n}}$

Page No.
Date :

4. ~~for~~ ($i = 1; i < n; i = i * 2$)
{ stet }

i
1
2
2^2
2^3
:
2^k

Assume $i >= n$
$\therefore i = 2^k$
$\therefore 2^k >= n$
$\therefore 2^k = n$
$\Rightarrow \log_2 2^k = \log_2 n$
$\therefore k = \log_2 n$

$\Rightarrow O(\log_2 n)$

5. $p = 0$
for ($i = 1; i < n; i = i * 2$)
{ p++; — $\log n$
}
for ($j = 1; j < p; j = j * 2$)
{ stet — $\log p$
}

$\Rightarrow O(\log \log n)$

6. for ($i = 0; i < n; i++$) — $n + n$
{ for ($j = 1; j < n; j = j * 2$) — $n(\log n)$
{
stet — $n(\log n)$
}
}

$2n \log n + n$
$O(n \log n)$

7. $i = 0;$ — 1
while (i < n) — $n+1$
{ stat; — n
i++; — n
} — $3n+2$

$\Rightarrow O(n)$

8. $a = 1;$
while (a < b)
{ stat;
a = a * 2;
}

a	Terminate
1	$a \geq b$
2	$2^k \geq b$
2^2	$2^k = b$
2^3	
⋮	
2^k	$\log_2 b = k$

$\Rightarrow O(\log_2 n)$

9. $i = n;$
while (i > 1)
{ stat;
$i = i/2;$
}

10. $i = 1;$
$k = 1;$
while (k < n)
{ stat
k = k + 1;
i++;
}

i	k
1	1
2	$1+1=2$
3	$2+2$
4	$2+2+3$
5	$2+2+3+4$
⋮	⋮
	$2+2+3+4+\dots m$
m	$\frac{m(m+1)}{2}$

$k \geq n$

$\frac{m(m+1)}{2} \geq n$

$m^2 \geq n$

$m \geq \sqrt{n}$

$\Rightarrow \boxed{O(\sqrt{n})}$

5	6	2	1	4

i=0, i=1, j, j+1, 6 > 2, 6, temp j


```
while (m != n)
  if (m > n)
    m = m - n;
  else
    n = n - m;
```

m	n
m = 16	n = 2
14	2
12	2
10	2
8	2
6	2
4	2
2	2

$\frac{16}{2}$

$\frac{n}{2}$

$\Rightarrow O(n)$

* bubble sort(a)

	cost	#
{		
for i=0 to n-1	c_1	n
for j=i to n	c_2	$n(n-1)$
if a[j] > a[j+1]	c_3	$n(n-2)$
temp = a[j]	c_4	$n(n-2)$
a[j] = a[j+1]	c_5	$n(n-2)$
a[j+1] = temp	c_6	$n(n-2)$

$T(n) = c_1 n + n(n-1)c_2 + n(n-2)c_3 +$

Page No.
Date :

```
* for (i=0; c1++; i<n-1; i++)        c1    n+1
{
    swap=0;                          c2    n
    for(j=0; c2++ ; j<n-i-1; j++)    c3    n(n)
    {
      if(arr[j] > arr[j+1])          c4    n(n-1)
      {
        swap++;                      c5    n(n-1)
        temp = arr[j];               c6    n(n-1)
        arr[j] = arr[j+1];           c7    n(n-1)
        arr[j+1] = temp;             c8    n(n-1)
      }
    }
    if(swap==0)                      c9    n
        break;
}
```

$n=1$ Best case

$$T(n) = 2c_1 + c_2 + n c_3 + (n-1)c_4 + (n-1)c_5 + (n-1)c_6 + (n-1)c_7 + (n-1)c_8 + c_9$$

$$= (2c_1 + c_2 + c_4 - c_5 - c_6 - c_7 - c_8 + c_9) + n(c_3 + c_4 + c_5 + c_6 + c_7 + c_8)$$

$$= an + b$$

$$T(n) = O(n)$$

1 2 4 5 key

Page No.
Date :

| 2 | 5 | 4 |

i j key = 2

* insertion_sort (A)

for j=2 to A.length	c_1	n
{	c_2	
key = A[j]	c_2	$n-1$
i = j-1	c_3	$n-1$
while (i>0 && A[i] > key)	c_4	$\sum_{j=2}^{n} t_j$
{		
A[i+1] = A[i]	c_5	$\sum_{j=2}^{n} (t_j-1)$
i = i-1	c_6	$\sum_{j=2}^{n} (t_j-1)$
}		
A[i+1] = key	c_7	$(n-1)$
}		

$$T(n) = n c_1 + (n-1)c_2 + (n-1)c_3 + c_4\left(\sum_{j=2}^{n} t_j\right) + \left(\sum_{j=2}^{n}(t_j-1)\right)c_5 + \sum_{j=2}^{n}(t_j-1)\,c_6 + (n-1)c_7$$

Best case $j = 2, 3, \dots n$ $t_j = 1$

$$T(n) = c_1 n + c_2 n - c_2 + c_4 \sum_{j=2}^{n} 1 + c_3(n-1) + c_5 \sum_{j=2}^{n}(1-1) + c_6 \sum_{j=2}^{n}(1-1) + n c_7 - c_7$$

$$= (\cancel{c_1} + c_2 + c_4)n * c_{\cancel{3}} \cancel{+ c_2} + \cancel{c_7} + 2$$

$$= c_1 n + c_2 n - c_2 + c_4(n-2) + c_7(n) - c_7 + n c_3 - c_3$$

$$= (c_1 + c_3 + c_2 + c_7 + c_4)n + (-c_2 - 2 - c_7 - c_3)$$

$$= an + b$$

$$T(n) \approx O(n)$$

AP = total no. of elements $\left(\frac{\text{first} + \text{last}}{2}\right)$ avg.

Page No. t_1
Date :

Worst case $t_j = j$, $j = 2, 3, \ldots n$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} j + c_5 \sum_{j=2}^{n} (j-1) + c_6 \sum_{j=2}^{n} (j-1) + c_7(n-1)$$

$$\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1$$

$$T(n) = n(c_1 + c_2 + c_3 + c_7) - (c_2 + c_3 + c_7) + c_4\left(\frac{n(n+1)}{2} - 1\right) + c_5\left(\frac{n(n+1)}{2} - 1\right)$$
$$- c_5 \sum_{j=2}^{n} 1 + c_6\left(\frac{n(n+1)}{2} - 1\right) + \left(-c_6 \sum_{j=2}^{n} 1\right)$$

$$= n(c_1 + c_2 + c_3 + c_7) - (c_2 + c_3 + c_7) + \left[\frac{n^2 + n - 2}{2} - (n-1)\right] c_5 +$$
$$c_4\left[\frac{n^2 + n - 2}{2}\right] + c_6\left[\frac{n^2 + n - 2}{2} - (n-1)\right]$$
$$+ \cancel{c_7(n)}$$

$$= n(c_1 + c_2 + c_3 + c_7) - (c_2 + c_3 + c_7) + \left(\frac{n^2 - n}{2}\right) c_5 + c_6\left(\frac{n^2 - n}{2}\right)$$
$$+ \left(\frac{n^2 + n - 2}{2}\right) c_4$$

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

Master theorem

Page No.
Date :

$T(n) = af(n/b) + f(n)$; $a \geq 1, b > 1$

compare $n^{\log_b a}$ with $f(n)$

* **case 1 :-** $f(n) = O(n^{\log_b a})$

solution is, $T(n) = \Theta(n^{\log_b a})$

1) $T(n) = 9T(n/3) + n$

$a = 9, b = 3, f(n) = n$

compare $n^{\log_b a}$

$n^{\log_3 3^2}$

n^2

$F(n) = O(n^{\log_b a})$

$F(n) \leq c \cdot (n^2)$

$n \leq c \cdot n^2$

As per applying case 1

$T(n) = \Theta(n^{\log_b a})$

$T(n) = \Theta(n^2)$

For case 2 $f(n) = g(n)$

Page No.
Date :

* Case - 2 :- $f(n) = \Theta(n^{\log_b a})$

solution : $T(n) = \Theta(n^{\log_b a} \log n)$

2) $T(n) = T(2n/3) + 1$

$a = 1$, $b = 3/2$, $f(n) = 1$

compare $n^{\log_b a}$

$= n^{\log_{3/2} 1}$

$= n^0$

$n^{\log_b a} = 1$ $f(n) = 1$

For applying case:2

$T(n) = \Theta(n^{\log_b a} \cdot \log n)$

$= \Theta(\log n)$

* Case:3 $F(n) = \Omega(n^{\log_b a})$

and $af(n/b) \leq c \cdot f(n)$

For some constant $0 \leq c \leq 1$

→ regularity condition

Solution : $T(n) = \Theta f(n)$

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

$a = 2$, $b = 2$, $f(n) = n^2$

$n^{\log_b a} = n^{\log_2 2} = n$

Page No.
Date :

$f(n) = n^2$
$n^2 \geq n$
$f(n) \neq \Omega(n^{\log_b a})$
$f(n) = n^2, b = 2$
$f(n/b) = n^2/4$

$a f(n/b) = 2(n^2/4)$
$= n^2/2$

$2(n^2/4) \leq c \cdot n^2$
$n^2/2 \leq c \cdot n^2$ $\quad 0 \leq c \leq 1$
regularity condit

$c = 1/2$

Applying case 3 $\quad T(n) = \Theta(n^2)$

ex. $T(n) = 2T(n/2) + \sqrt{n}$
$a = 2, b = 2, f(n) = n^{1/2}$

compare $n^{\log_b a} = n^{\log_2 2}$
$= n$

$f(n) = O(n)$
$n^{1/2} \leq c.(n)$
Applying case: 1
$T(n) = \Theta(n^{\log_b a})$

ex ~~ex~~ ~~case - 2~~ f.

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

$a = 2^n$, $b = 2$, $f(n) = n^n$

$n^{\log_b a} = n^{\log_2 2^n} = n^n$

We can not solve master theorem because here here a is exponential cond value.

ex $T(n) = 2T(n/2) + n^4$

$a = 2$, $b = 2$, $f(n) = n^4$

$n^{\log_b a} = n^{\log_2 2}$

$= n$

case 3 : $f(n) \geq n^{\log_b a}$

$n^4 \geq n$

$f(n) = \Omega(n^{\log_b a})$ $f(n) = n^4$

$af(n/b) \leq c \cdot n^4$ $f(n/2) = (n/2)^4$

$2f(n/2) \leq c \cdot n^4$ $= n^4/16$

$2\left(\frac{n^4}{16}\right) \leq c \cdot n^4$

$\frac{1}{8} = c$

$c = 1/8$

$T(n) = \Theta(n^4)$

ex. $\cancel{T(n) = 2T(n/2) + n^2}$

$\cancel{a = 2, \quad b = 2, \quad f(n) = n^2}$

ex. $T(n) = 3T(n/4) + n\log n$

$a = 3, \quad b = 4, \quad f(n) = n\log n$

$n^{\log_b a} = n^{\log_4 3}$

$=$

$f(n) > n^{\log_4 3}$

$f(n) = \Omega(n^{\log_4 3})$

$f(n/4) = (n/4)\log(n/4)$

$af(n/4) = 3(n/4)\log(n/4)$

$3\left(\frac{n}{4}\right)\log(n/4) \le c \cdot n\log n$

$3/4\, n\log n/4 \le 3/4\, n\log n$

$\boxed{c = 3/4}$

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

ex. $T(n) = 4T(n/3) + n\log n$

$a = 4, \quad b = 3$

$n^{\log_b a} = n^{\log_3 4 > 1}$ $\quad f(n) = n\log n$

$f(n) < n^{\log_3 4}$

$f(n) = O(n^{\log_3 4})$

case 1 :-

$T(n) = \Theta(n^{\log_3 4})$

Page No.
Date :

ex: $T(n) = 4T(n/2) + (n^2\sqrt{n})$

$a = 4$, $b = 2$

$n^{\log_b a} = n^{\log_2 2^2} = n^2$

$f(n) > n^{\log_b a}$

$f(n) = \Omega(n^{\log_b a})$ case 3

$T(n) = \Theta(n^2\sqrt{n})$

$a f\left(\frac{n}{2}\right)^2 \sqrt{\frac{n}{2}} \leq c \cdot n^2\sqrt{n}$

$\cancel{4}\frac{n^2}{\cancel{4}}\frac{\sqrt{n}}{\sqrt{2}} \leq c \cdot n^2\sqrt{n}$

$c = 1/\sqrt{2}$

$T(n) = \Theta(n^2\sqrt{n})$

ex: $T(n) = 0.5\, T(n/2) + n$

$a = 1/2$, $b = 2$

Here a is less than 1. So, it is not
we can not apply master theorem.

ex:

$T(n) = 64\, T(n/8) - n^2 \log n$

$a = 64$ $b = 8$ $f(n) = -n^2 \log n$

$n^{\log_b a} = n^{\log_8 8^2}$

$= n^2$

we can not apply master theorem, because $f(n)$ is negative.

ex: $T(n) = T(n/2) + n(2 - \cos n)$

$a = 1$, $b = 2$, $f(n) = n(2 - \cos n)$

we can not apply master theorem, because $f(n)$ is cosine function.

ex: $T(n) = 7T(n/2) + \Theta(n^2)$

$a = 7$ $b = 2$ $f(n) = n^2$

$n^{\log_b a} = n^{\log_2 7 > 2}$

$f(n) \leq n^{\log_2 7}$

$f(n) = O(n^{\log_2 7})$

$T(n) = \Theta(n^{\log_2 7})$

ex: $T(n) = cn + 3T(2n/3)$

$a = 3$, $b = 3/2$ $f(n) = c \cdot n$

$n^{\log_b a} = n^{\log_{3/2} 3}$

8. $T(n) = 2T(n/2) + c \cdot n$

$a = 2, b = 2$

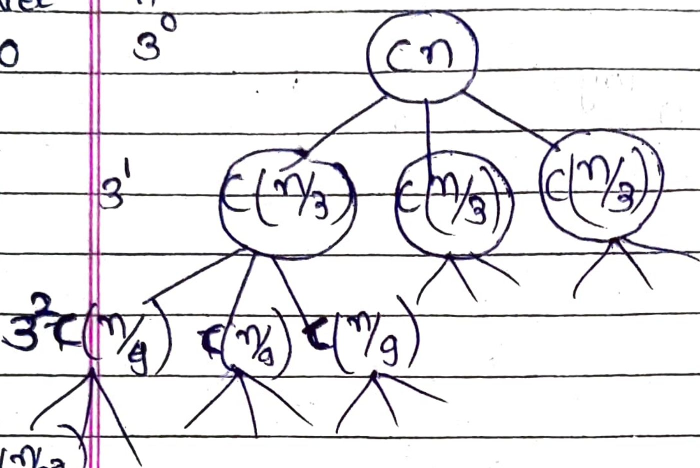
$n^{\log_b a} = n^{\log_2 2} = n$

(1.) $T(n) = 3T(n/3) + c(n)$

$T(n/3) = 3T(n/9) + c(n/3)$

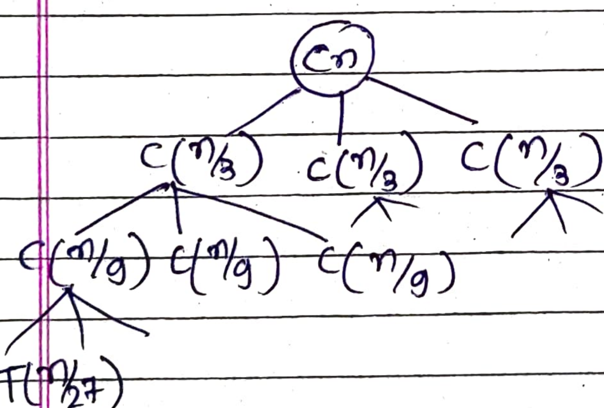
$T(n/9) = 3T(n/27) + c(n/9)$

level	#
0	3^0
1	3^1
2	3^2
3	3^3
i	3^i
h	3^h



T(1) ... T(1) .... T(1)

(where we get best case)



level	sum
0	$3^0 \times c(n) = cn$
1	$3^1 c(n/3) = cn$
2	$9 \cdot c(n/9) = cn$
⋮	
h-1	$3^{h-1} c \cdot (n/3^{h-1}) = cn$
h	$3^h T(1)$

$$\frac{n}{3^h} = 1$$

$$n = 3^h$$

$$\log_3 n = \log_3 3^h$$

$$h = \log_3 n$$

$$T(n) = 3^h\, T(1) + \sum_{i=0}^{h-1} cn$$

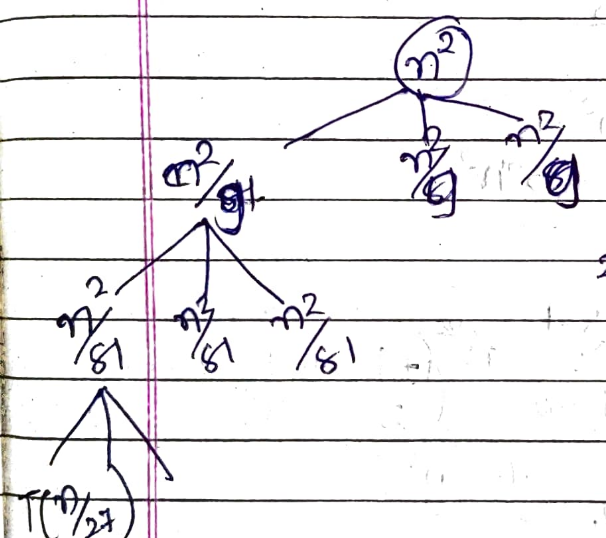
$$= 3^h\, T(1) + h \cdot c \cdot n$$

$$= 3^{\log_3 n}\, T(1) + \log_3 n \cdot c \cdot n$$

$$= n + n(\log_3 n) \cdot c$$

$$T(n) \approx O(n \log_3 n)$$

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



0	3^0	$T(n/3) = 3T(n/9) + n^2/9$
1	3^1	$T(n/9) = 3T(n/27) + n^2/81$
2	3^2	

level	sum (no. of node generated × size)
0	$3^0 \times n^2 = n^2$
1	$3^1 \times \frac{n^2}{3^2} = \frac{n^2}{3}$
2	$3^2 \times \frac{n^2}{3^4} = \frac{n^2}{3^2}$
⋮	
$h-1$	$3^{h-1} \times \frac{n^2}{3^{(h-1)^2}} = \frac{n^2}{3^{h-1}}$
h	$3^h \times T(1) = \frac{n^2}{3^h}$

$$\frac{n^2}{3^h} = 1$$

$$n^2 = 3^h$$

$$\log_3 n^2 = \log_3 3^h$$

$$h = \log_3 n^2$$

$$T(n) = nT(1) + n^2\left(\frac{n-1}{n}\right)\frac{3}{2}$$

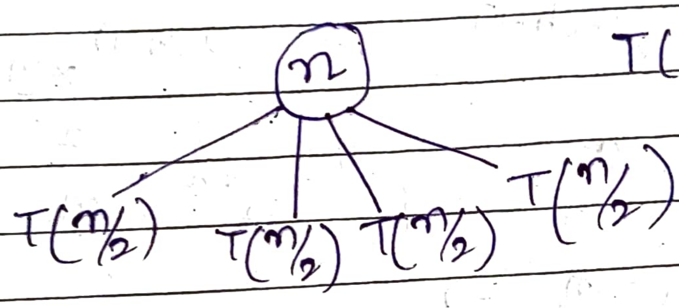
$$= nT(1) + \frac{3n(n-1)}{2}$$

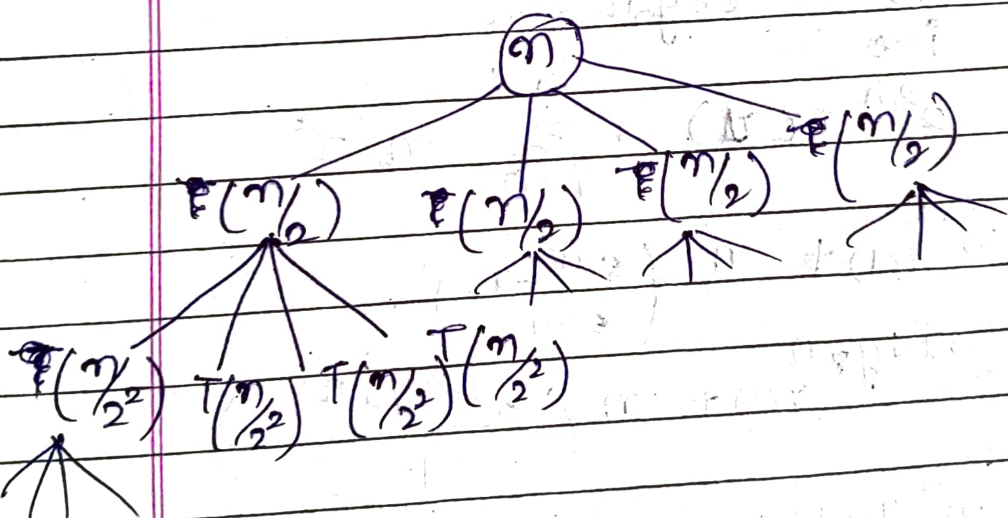
$$= n \cdot T(1) + \frac{3}{2}n^2 - \frac{3}{2}n$$

$$T(n) \approx O(n^2)$$

(3) $T(n) = 4T(n/2) + n$

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





level	#
0	1
1	4
2	4^2
⋮	
i	4^i
⋮	
$h-1$	4^{h-1}
h	4^h

Level	sum
0	$1 \times n/\cancel{2} = n/\cancel{2}$
1	$2^2 \times \frac{n}{2} = 2n$
2	$2^4 \times \frac{n}{2^2} = 2^2 n$

Page No. size of subpr
Date :

level	#	sum
0	1	$1 \times n = n$
1	2^2	$2^2 \times \frac{n}{2} = 2n$
2	2^4	$2^4 \times \frac{n}{2^2} = 2^2 n$
i	2^{2i}	$2^{2i} \times \frac{n}{2^i} = 2^i n$
$h-1$	$2^{2(h-1)}$	$2^{2(h-1)} \times \frac{n}{2^{h-1}} = 2^{h-1} n$
h	2^{2h}	$2^{2h} \times \frac{n}{2^h} = 2^h n$

height of tree :- $\frac{n}{2^h} = 1$

$T(n) = \cancel{\Sigma}_{i=0}$ $\quad \log_2 n = h$

$$T(n) = 2^{2h} T(1) + \sum_{i=0}^{h-1} n 2^i$$

$$= 2^{2h} T(1) + n\left(\frac{2^h - 1}{2-1}\right)$$

$$= 2^{2\log_2 n} T(1) + n\left(\frac{2^{\log_2 n} - 1}{1}\right)$$

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

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

$$T(n) = c_1(n+1) + nc_2 + n^2c_3 + n^2c_4 - \cancel{n}c_4$$
$$+ n^2c_5 - nc_5 + n^2c_6 - nc_6 +$$
$$n^2c_8 - nc_8 + n^2c_7 - nc_7 + nc_9$$

For best case, $n=1$

$T(n) = t_1(2) + \bar{t}\cdots$

$= 2c_1 + c_2 +$

* $T(1) = 1$

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

Page No.
Date :

	cost	#
• bubble_sort (a)		
{		
for i=0 to n-1	c_1	n
for j=i to n	c_2	n(n-1)
if a[j] > a[j+1]	c_3	n(n-2)
temp = a[j]	c_4	n(n-2)
a[j] = a[j+1]	c_5	n(n-2)
a[j+1] = temp	c_6	n(n-2)

$$T(n) = nc_1 + n^2c_2 - \cancel{n}c_2 + n^2c_3 - 2c_3 + n^2c_6 - 2nc_6 + n^2c_4 - 2c_4 + n^2c_5 - 2c_5$$

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

	cost	#
• for (i=0 ; c1++, i<n ; i++)	c_1	n+1
{ swap = 0 ;	c_2	n
for (j=0 ; c2++ , j<n-i-1 ; j++)	c_3	n(n)
{		
if (arr[j] > arr[j+1])	c_4	n(n-1)
{		
swap++ ;	c_5	n(n-1)
temp = arr[j] ;	c_6	n(n-1)
arr[j] = arr[j+1] ;	c_7	n(n-1)
arr[j+1] = temp;	c_8	n(n-1)
}		
}		
if (swap==0)	c_9	n
break;		
}		

Page No.
Date :

☆ Summation formula and rules used in efficiency analysis

1) $\sum_{i=1}^{n} 1 = 1+1+\cdots n \approx \in O(n)$

2) $\sum_{i=1}^{n} i = 1+2+3+\cdots+n = \in O\frac{(n)(n+1)}{2}$

$\in O(n^2)$

3) $\sum_{i=1}^{n} i^k = 1+2^k+3^k+\cdots n^k = \in O(\frac{n^{k+1}}{k+1})$

$= \in O(n^k)$

4) $\sum_{i=1}^{n} a^i = a+a^2+a^3+\cdots a^n \approx \frac{a^{n+1}-1}{a-1}$

$\in O(a^n)$

5) $\sum_{i=1}^{n}(a_i \pm b_i) \approx \sum_{i=1}^{n} a_i \pm \sum_{i=1}^{n} b_i$

6) $\sum_{i=1}^{n} c\,a_i = c\sum_{i=1}^{n} a_i$

7) $\sum_{i=k}^{n} = n-k+1$

8) $\sum_{i=0}^{h} \frac{1}{a^i} = \frac{a(1-r^k)}{(1-r)}$ $\quad r<1$

$a^{\log_a P} = P$

3. $T(0) = c_1$ $\quad T(1) = c_2$

$T(n) = 2T(n/2) + c_3$ $\quad T(n/2) = 2T(n/4) + c_3$

$T(n) = 2T(n/4) + 2c_3 + c_3$

$= 4T(n/4) + 3c_3$

$T(n) = 8T(n/8) + 3c_3$ $\quad \frac{n}{2^k} = 1$

$= 2^k T(n/2^k) + k c_3$ $\quad n = 2^k$

$= 2^k T(1) + \log_2 n \, c_3$ $\quad \log_2 n = \log_2 2^k$

$= 2^k c_2 + (\log_2 n) c_3$ $\quad \log_2 n = k$

$= n c_2 + (\log_2 n) c_3$

$T(n) = 8T(n/4) + 6c_3 + c_3$

$= 8T(n/8) + 7c_3$

$= 2^k T(n/2^k) + (2^k - 1) c_3$

$= \cancel{2^n T(n/2^n) + (2^k - 1) c_3}$

$= \cancel{2^n T(1) + \log_2 n k - c_3}$

$= \cancel{2^n c_2 +}$

$= 2^k T(n/2^k) + (2^k - 1) c_3$

$= 2^k T(1) + (n) c_3 - c_3$

$= 2^{\log_2 n} T(1) + (n-1) c_3$

$= n c_2 + (n-1) c_3$

$T(n) = O(n)$

Substitution method

Substitution method

(1) $T(0) = C_1$

$T(N) = C_2 + T(n-1)$

$T(n) = T(n-1) + C_2$ (put $n = n-1$)

$T(n-1) = T(n-2) + C_2$

$T(n) = C_2 + T(n-2) + C_2$ $\quad T(n-2) = T(n-2-1) + C_2$

$T(n) = 2C_2 + T(n-2)$

$T(n) = 2C_2 + T(n-3) + C_2$ $\quad T(n-2) = T(n-3) + C_2$

$T(n) = T(n-3) + 3C_2$

for $n = K$

$T(n) = T(n-K) + KC_2$

$T(n) = T(0) + nC_2$ $\quad n = K$

$= C_1 + nC_2$

$T(n) = O(n)$

2. $T(0) = C_1$

$T(1) = C_2$

$T(n) = T(n/2) + C_3$

$T(n) = T(n/4) + C_3 + C_3$ $\quad T(n) = T(n/2) + C_3$

$= T(n/4) + 2C_3$ $\quad T(n/2) = T(n/4) + C_3$

$T(n) = T(n/8) + 3C_3$ $\quad T(n/4) = T(n/8) + C_3$

$T(n) = T(n/2^K) + KC_3$

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

~~$n = K$~~

$T(n) = T(n/2^n) + nC_3$ $\quad n = 2^k$

$T(n) = T(1) + \log(n) C_3$ $\quad \log_2 n = \log_2 2^k$

$= C_2 + (\log n) C_3$ $\quad \log_2 n = K$

$T(n) = O(\log_2 n)$