Time Analysis

1. Swap (a,b){

temp = a ———— 1

a = b ———— 1

b = temp ———— 1

}

$\therefore f(n) = 3$

$\therefore O(1)$ [constant]

Space

a —— 1

b —— 1

temp —— 1

$S(n) = 3$

$\therefore O(1)$

2. Void sum(A[ ], n){

s = 0 ———— 1

for (i=0; i<n; i++){ ———— n+1

S = S + A[i] ———— n

}

}

$f(n) = n + n + 1 + 1$
$= 2n + 2$

$\therefore O(n)$

A[ ] —— n

n —— 1

i —— 1

s —— 1

$\therefore S(n) = n + 3$

$\therefore O(n)$

3. Add (A, B, n) {

n+1 ———— for (i=0; i<n; i++) {

n(n+1) ———— for (j=0; j<n, j++){

n × n ———— C[i,j] = A[i,j] + B[i,j]

}

}

$f(n) = n + 1 + n^2 + n + n^2$
$= 2n^2 + 2n + 1$

$\therefore O(n^2)$

A —— n^2

B —— n^2

n —— 1

i —— 1

j —— 1

C —— n^2

$S(n) = 3n^2 + 3$

$\therefore O(n^2)$

4. for(i=0; i<n; i++){ —— $n+1$
statement —— n
}

for(i=0; i<n; i++){ —— $n+1$
statement —— n
}

$f(n) = n+1+n+n+1+n$
$= 4n+2$
$\therefore O(n)$

i — 1
n — 1

$s(n) = 2$
$\therefore O(1)$

5. for (i=n; i>0; i--){ —— $n+1$
statement —— n
}

$f(n) = n+1+n$
$= 2n+1$
$\therefore O(n)$

i — 1
n — 1

$s(n) = 2$
$\therefore O(1)$

6. for (i=1; i<n; i=i+2) {
statement —— $\frac{n}{2}$
}

$O(n)$

7. for (i=0; i<n; i++) { —— $n+1$
for (j=0; j<n; j++) { —— $n(n+1)$
statement —— n^2
}
}

$f(n) = n+1+n^2+n+n^2$
$= 2n^2+2n+1$
$\therefore O(n^2)$

i — 1
j — 1
n — 1

$s(n) = 3$
$\therefore O(1)$

8.
```
for (i=0; i<n; i++){
    for (j=0; j<i; j++){
        statements;
    }
}
```

i	j	no. of time
0	0	0
1	0 1	1 0 } 1
2	0 1 2	1 1 0 } 2
⋮		
n		n

$\therefore 1+2+\dots+n = \frac{n(n+1)}{2}$

$\therefore f(n) = \frac{n^2+n}{2}$

$\therefore O(n^2)$

9.
```
p=0;
for(i=1; p<=n; i++){
    p = p+i;
}
```

i	p
1	0+1
2	0+1+2
3	0+1+2+3
⋮	
k	$0+1+2+3+\dots+k = \frac{k(k+1)}{2}$

let, $p > n$

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

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

$\Rightarrow k^2 > n$

$\therefore k > \sqrt{n}$ $\therefore O(\sqrt{n})$

Scanned with CamScanner

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

{statement}

i	
1	$= 2^0$
1×2	$= 2$
$1\times2\times2$	$= 2^2$
$1\times2\times2\times2$	$= 2^3$
$\vdots$	
2^k	

Assume, $i > n$

$\Rightarrow 2^k > n$

$\therefore k > \log_2 n$

$\therefore O(\log_2 n)$

11. for(i=n; i>=1; i=i/2) {

start

}

i	
n	
$\frac{n}{2}$	
$\frac{n}{2\times2}$	$= \frac{n}{2^2}$
$\frac{n}{2\times2\times2}$	$= \frac{n}{2^3}$
$\vdots$	
$\frac{n}{2^k}$	

Assume,

$i < 1$

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

$\Rightarrow n < 2^k$

$\therefore k > \log_2 n$

$\therefore O(\log_2 n)$

Scanned with CamScanner

12. for(i=0; i*i<n; i++){

start ———————— n

}

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

start ———————— n

}

$\therefore f(n) = n + n$

$= 2n$

$\therefore O(n)$

13. p=0

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

p++; ———————— $\log_2 n$

}

for(j=1; j<p; j=j*2){

start ———————— $\log_2 p$

}

$p = \log_2 n$

$\therefore O(\log \log n)$

Scanned with CamScanner

14.

```
for(i=0; i<n; i++){            ———— n+1
    for(j=1; j<n; j=j*2){      ———— (n+1) log_2 n
        start;                 ———— n x log_2 n
    }
}
```

$$f(n) = n+1 + \log_2 n + n\log_2 n + n\log_2 n$$
$$= 2n\log_2 n + \log_2 n + n + 1$$

$\therefore O(n\log n)$

15.

```
i=1
k=1
while (k<n) {
    start
    k=k+i
    i++
}
```

i	k
1	1
2	1+2
3	1+2+3
⋮	
m	1+2+3+ ⋯ +m

$$= \frac{m(m+1)}{2}$$

Assume,

$$k > n$$

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

$$\Rightarrow \frac{m^2+m}{2} > n$$

$$\Rightarrow m^2 > n$$

$$\therefore m > \sqrt{n}$$

$$\therefore O(\sqrt{n})$$

16. while (m != n) {

if (m > n)

m = m - n

else

n = n - m

}

	m	n	
(let)	6	3	2 $\therefore O(1)$ Best case
	3	3	
(let)	16	2	n $\therefore O(n)$ worst case
	14	2	
	12	2	
	⋮		
	2	2	

Asymptotic notation

$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 < \cdots < 2^n < 3^n \cdots < n^n$

- Big-oh (O) upperbound $\rightarrow$ (worst case)
- Big-Omega (Ω) Lower bound $\rightarrow$ (Best case)
- Theta (Θ) Average bound $\rightarrow$ (Avg case)

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

$$2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2$$

$$\Rightarrow 2n^2 + 3n + 4 \leq 9n^2 \qquad n \geq 1$$

$$\therefore f(n) = O(n^2)$$

$$2n^2 + 3n + 4 \geq 1 \times n^2$$

$$\therefore f(n) = \Omega(n^2)$$

$$n^2 \leq 2n^2 + 3n + 4 \leq 9n^2$$

$$\therefore f(n) = \Theta(n^2)$$

2.

3.

4.

2. $f(n) = n^2 \log n + n$

$n^2\log n \le n^2 \log n + n \le 10 n^2 \log n$

$\therefore O(n^2 \log n)$

$\Omega(n^2 \log n)$

$\theta(n^2 \log n)$

3. $f(n) = n!$

$= n \times (n-1) \times (n-2) \cdots \times 3 \times 2 \times 1$

$= 1 \times 2 \times 3 \times \cdots \times n$

$1 \times 1 \cdots \times 1 \le 1 \times 2 \times 3 \times \cdots \times n \le n \times n \times n \times \cdots \times n$

$1 \le n! \le n^n$

1 & n^n are not same, so we can't take θ.

$\Omega(1)$

$O(n^n)$

4. $f(n) = \log n!$

$\log(1 \times 1 \cdots \times 1) \le \log(1 \times 2 \times \cdots \times n) \le \log(n \times n \times \cdots \times n)$

$1 \le \log n! \le \log n^n$ or $n \log n$

$O(n \log n)$

$\Omega(1)$

Sorting Analysis & Simulation

Bubble Sort

```
BubbleSort (A[ ], n)
    for k←0 , n-1 do                      ——— n
        for i←0, n-2 do                   ——— (n-1) n
            if A[i] > A[i+1] then
                swap (A[i], A[i+1])       ——— n(n-1) (n-1)
            end if
        end for
    end for
end procedure
```

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

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

$\therefore O(n^2)$

1

| 20 | 9 | 6 | 3 | 1 |

| 9 | 20 | 6 | 3 | 1 |

| 9 | 6 | 20 | 3 | 1 |

| 9 | 6 | 3 | 20 | 1 |

2

| 9 | 6 | 3 | 1 | 20 |

| 6 | 9 | 3 | 1 | 20 |

| 6 | 3 | 9 | 1 | 20 |

| 6 | 3 | 1 | 9 | 20 |

3

| 6 | 3 | 1 | 9 | 20 |

| 3 | 6 | 1 | 9 | 20 |

| 3 | 1 | 6 | 9 | 20 |

| 3 | 1 | 6 | 9 | 20 |

4

| 3 | 1 | 6 | 9 | 20 |

| 1 | 3 | 6 | 9 | 20 |

sorted

Recursive Equation: $\boxed{T(n) = T(n-1) + n \quad \therefore T(n) = \begin{cases} 1 & ; n=0 \\ T(n-1)+n & ; n>0 \end{cases}}$

Substitution $\rightarrow$

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

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

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

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

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

$$\vdots$$

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

Assume,

$n-k=0$

$\therefore n=k$

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

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

$$\vdots$$

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

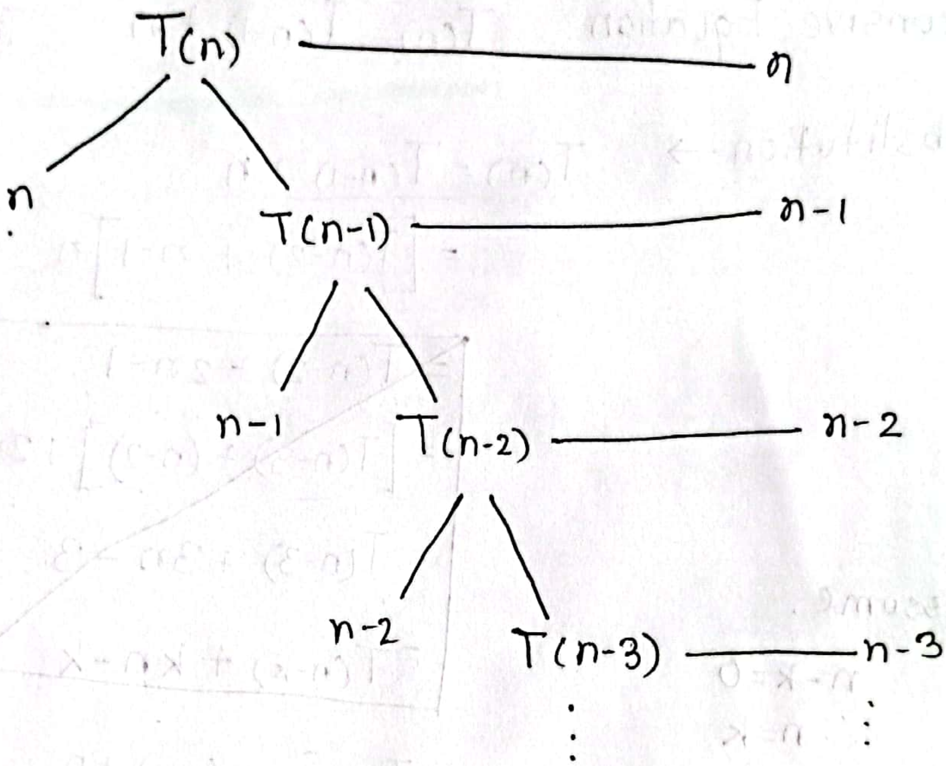
$$= T(n-n) + (n-n+1) + (n-n+2) + \cdots + (n-n) + n$$

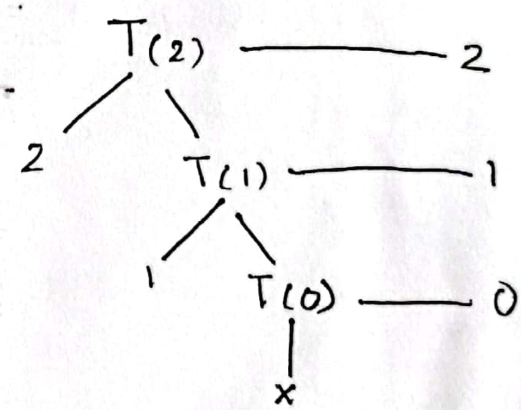
$$= T(0) + 1 + 2 + \cdots + (n-1) + n$$

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

$\therefore O(n^2)$

Tree →





$\therefore 0 + 1 + 2 + \dots + (n-3) + (n-2) + (n-1) + n$

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

$\therefore O(n^2)$

Insertion sort

```
InsertionSort(A[],n)
    for j←1 , n-1 do  ———————————  n
        value ← A[j]  ———————————  n-1
        i ← j-1  ———————————  n-1
        while i>=0 & A[i]>value do ———  Σ_j^n = 2 t_j
            swap (A[i] , A[i+1]) ———————  Σ_j^n = 2(t_j - 1)
            i ← i-1  ———————————  Σ_j^n = 2(t_j - 1)
        end while
        A [i+1] = value  ———————————  n-1
    end for
end procedure
```

$$f(n) = n + 3(n-1) + \sum_{j=2}^{n} t_j + 2\sum_{j=2}^{n}(t_j - 1)$$

Worst case: ~~O(n²)~~ [reversed]

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

$$= an^2 + bn + c$$

$\therefore O(n^2)$

$$\boxed{\text{Recursive: } T(n) = T(n-1) + n}$$

Best case: [sorted]

$$f(n) = n + 3(n-1) + n - 1$$

$$= an + b$$

$\therefore O(n)$

1 | 23 | 1 | 10 | 5 | 2 |

2 | 1 | 23 | 10 | 5 | 2 |

3 | 1 | 10 | 23 | 5 | 2 |

4 | 1 | 5 | 10 | 23 | 2 |

5 | 1 | 2 | 5 | 10 | 23 |

sorted

Selection sort

```
SelectionSort (A[ ] , n)
  for i = 1 to n-1 do                  ———— n
     min = i                           ———— (n-1)
     for j = i+1 to n  do              ———— (n-1)n
          if  A[j] < A[min] then       ———— (n-1)(n-1)
               min = j                 ———— (n-1)(n-1)
          end if
     end for
     Swap  A[min] and A[i]             ———— (n-1)
  end for
end procedure
```

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

$= 3n^2 - 2n$

$\therefore O(n^2)$

$\boxed{\text{Recursive Equation: } T(n) = T(n-1) + n}$

1	15	3	37	29	11
2	3	15	37	29	11
3	3	11	37	29	15
4	3	11	15	37	29
5	3	11	15	29	37

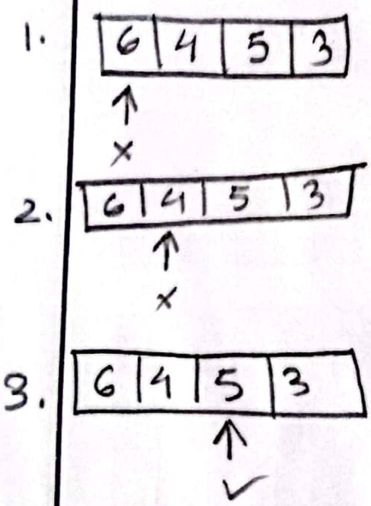
Linear Search

```
LinearSearch (A, n, item) .
    for i<-0 , n-1 do                 ______ n
        if A[i] == item then          ______ n-1
            return i                  ______ n-1
        end if
    end for
    return -1                         ______ 1
end procedure
```

$$f(n) = n + n-1 + n-1 + 1$$
$$= 3n+1$$

$O(n)$

item=5



Quick sort

```
void quickSort (Arr[ ], low, high) {                    (C1)
    if(low < high) { then
        (C2) pi = partition (Arr, low, high);           —— n
        1 —— quickSort(Arr, low, pi-1)                  —— n/2
        n-1 —— quickSort (Arr, pi+1, high);             —— n/2
    end } if
end } procedure

partion (Arr[ ], low, high)
        pivot = Arr[low]
        i = low
for (j=low+1 to j<=high) do
        if Arr[j]<pivot then
            i++
            swap(Arr[i], Arr[j])
        end if
end for
swap (Arr[i], Arr[low])
end procedure
```

(partition, mid)

Best case:

$O(n \log n)$

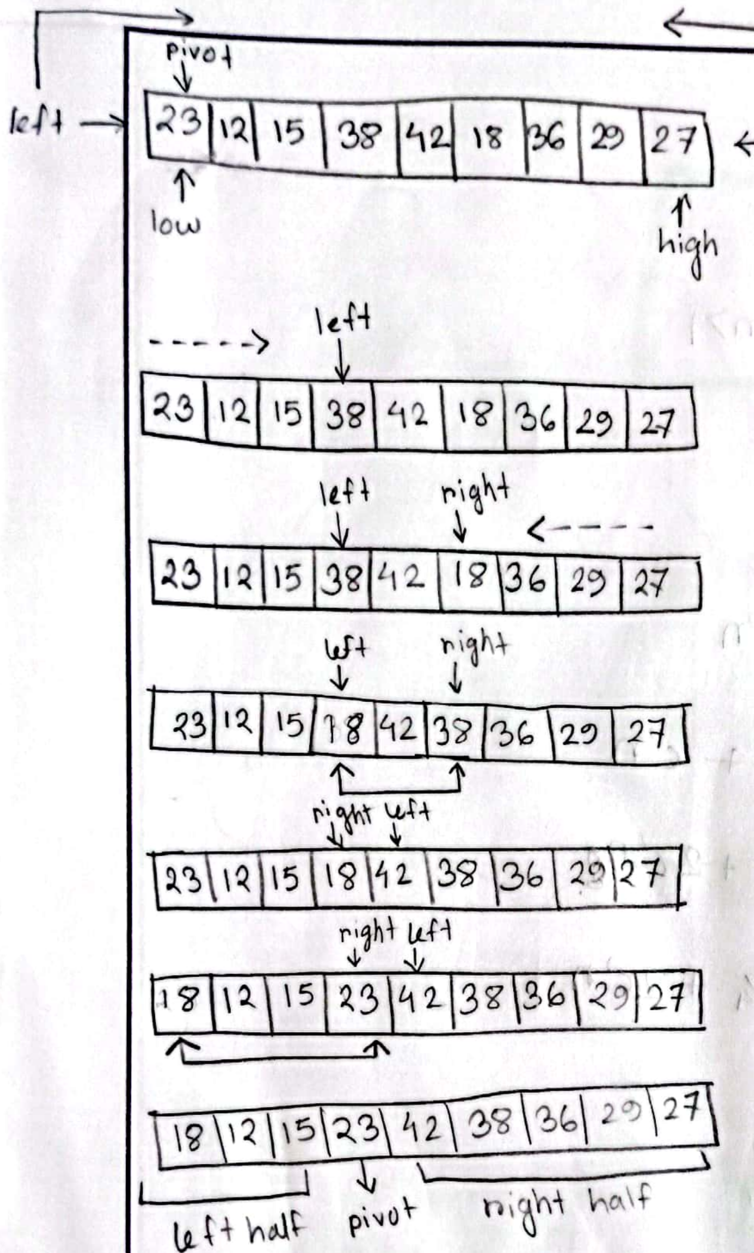
$\frac{n(n+1)}{2}$

Worst case

$O(n^2)$

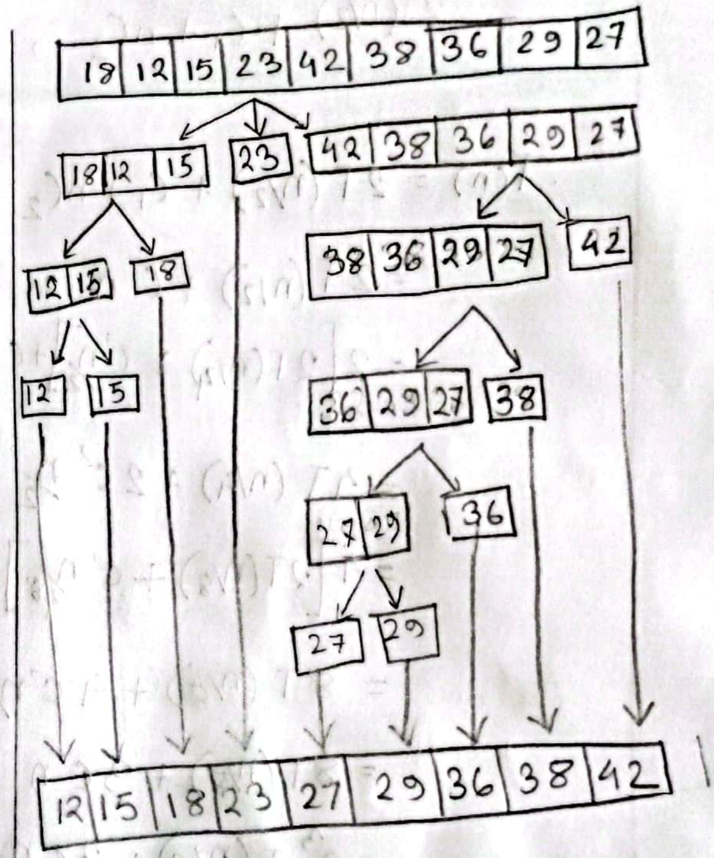
(already sorted)

left


pivot
left
23 | 12 | 15 | 38 | 42 | 18 | 36 | 29 | 27
right
low
high
left
23 | 12 | 15 | 38 | 42 | 18 | 36 | 29 | 27
left
right
23 | 12 | 15 | 38 | 42 | 18 | 36 | 29 | 27
left
right
23 | 12 | 15 | 18 | 42 | 38 | 36 | 29 | 27
right
left
23 | 12 | 15 | 18 | 42 | 38 | 36 | 29 | 27
right
left
18 | 12 | 15 | 23 | 42 | 38 | 36 | 29 | 27
18 | 12 | 15 | 23 | 42 | 38 | 36 | 29 | 27
left half
pivot
right half

left ++

right --


18 | 12 | 15 | 23 | 42 | 38 | 36 | 29 | 27
18 | 12 | 15
23
42 | 38 | 36 | 29 | 27
12 | 15
18
38 | 36 | 29 | 27
42
12
15
36 | 29 | 27
38
27 | 29
36
27
29
12 | 15 | 18 | 23 | 27 | 29 | 36 | 38 | 42

Recursive equation: (Best)

$$\boxed{T(n) = \begin{cases} O(1) & ; n=1 \\ 2T(n/2) + c_1 + nc_2 & ; n>1 \end{cases}}$$

$\therefore T(n) = 2T(n/2) + c_1 + nc_2$

$= 2T(n/2) + c'n$

$= 2\left[2T(n/4) + c'\frac{n}{2}\right] + c'n$

$= 4T(n/4) + 2c'\frac{n}{2} + c'n$

$= 4\left[2T(n/8) + c'\frac{n}{4}\right] + 2c'n$

$= 8T(n/8) + 4c'\frac{n}{4} + 2c'n$

$= 8T(n/8) + 3c'n$

$= 2^3 T(n/2^3) + 3c'n$

$\vdots$

$= 2^k T(n/2^k) + kc'n$

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

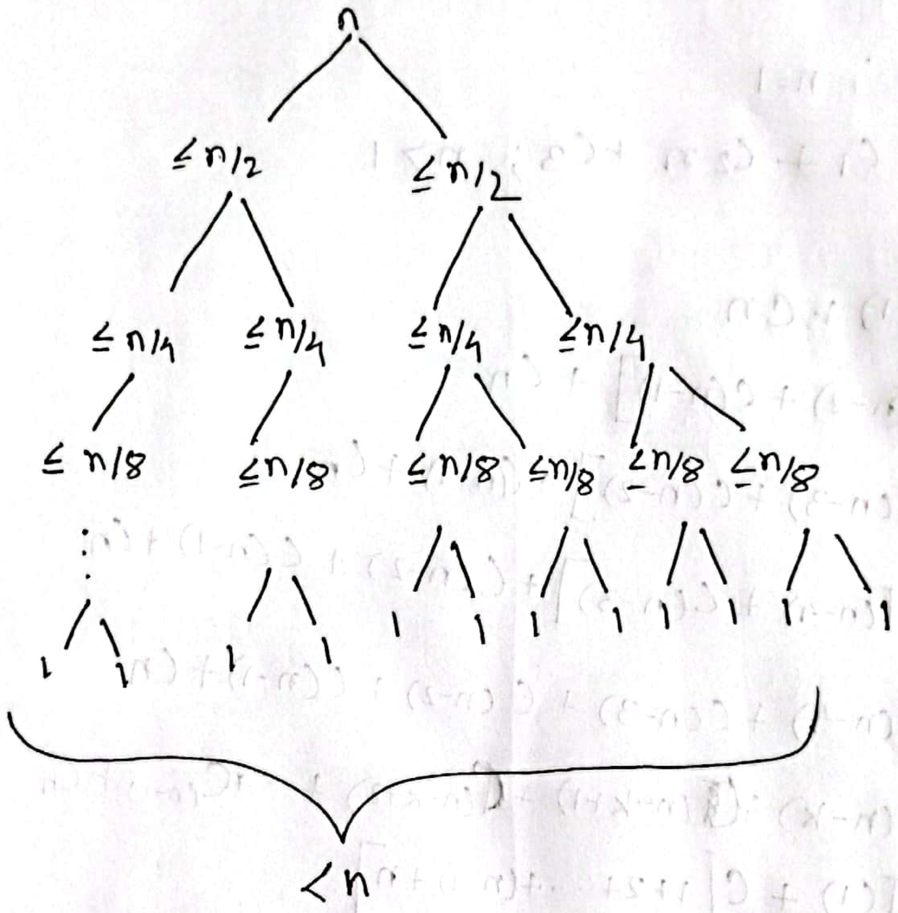
$\Rightarrow n = 2^k$

$\therefore k = \log_2 n$

$= 2^{\log_2 n} T(1) + c' n \log_2 n$

$= nC + c' n \log_2 n$

$\therefore O(n \log n)$



$\therefore$ cn

$\leq 2\,cn/2 = cn$

$\leq 4\,cn/4 = cn$

$\leq 8\,cn/8 = cn$

$\vdots$

$< nc = cn$

$\therefore O(\log_2 n)$

(worst)

$$T(n) = \begin{cases} O(1) & ; n=1 \\ T(n-1) + c_1 + c_2 n + c_3 & ; n>1 \end{cases}$$

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

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

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

$= [T(n-4) + c(n-3)] + c(n-2) + c(n-1) + cn$

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

$\vdots$

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

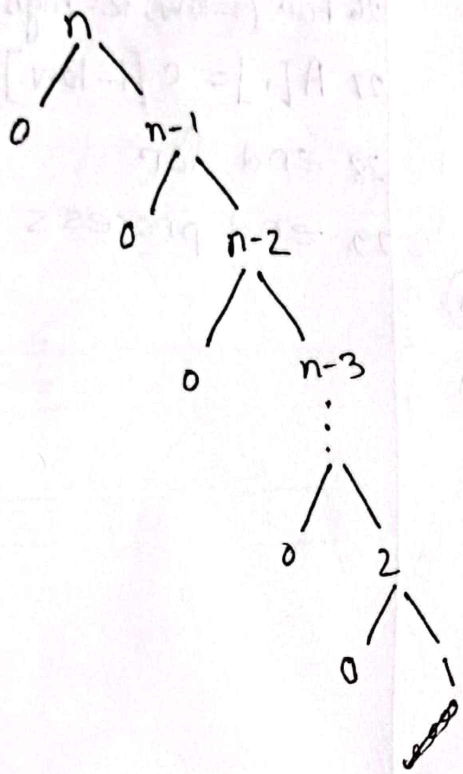
Assume, $k = n-1$

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

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

$\therefore O(n^2)$

Scanned with CamScanner



$\therefore$ cn

$c(n-1)$

$c(n-2)$

$c(n-3)$

$\vdots$

$2c$

c

0

$\therefore\ cn + c(n-1) + c(n-2) + c(n-3) + 2c + c \cancel{= c(n+n-1+n-2+n-3+2+1)}$

$\cancel{= c(4n-4)}$

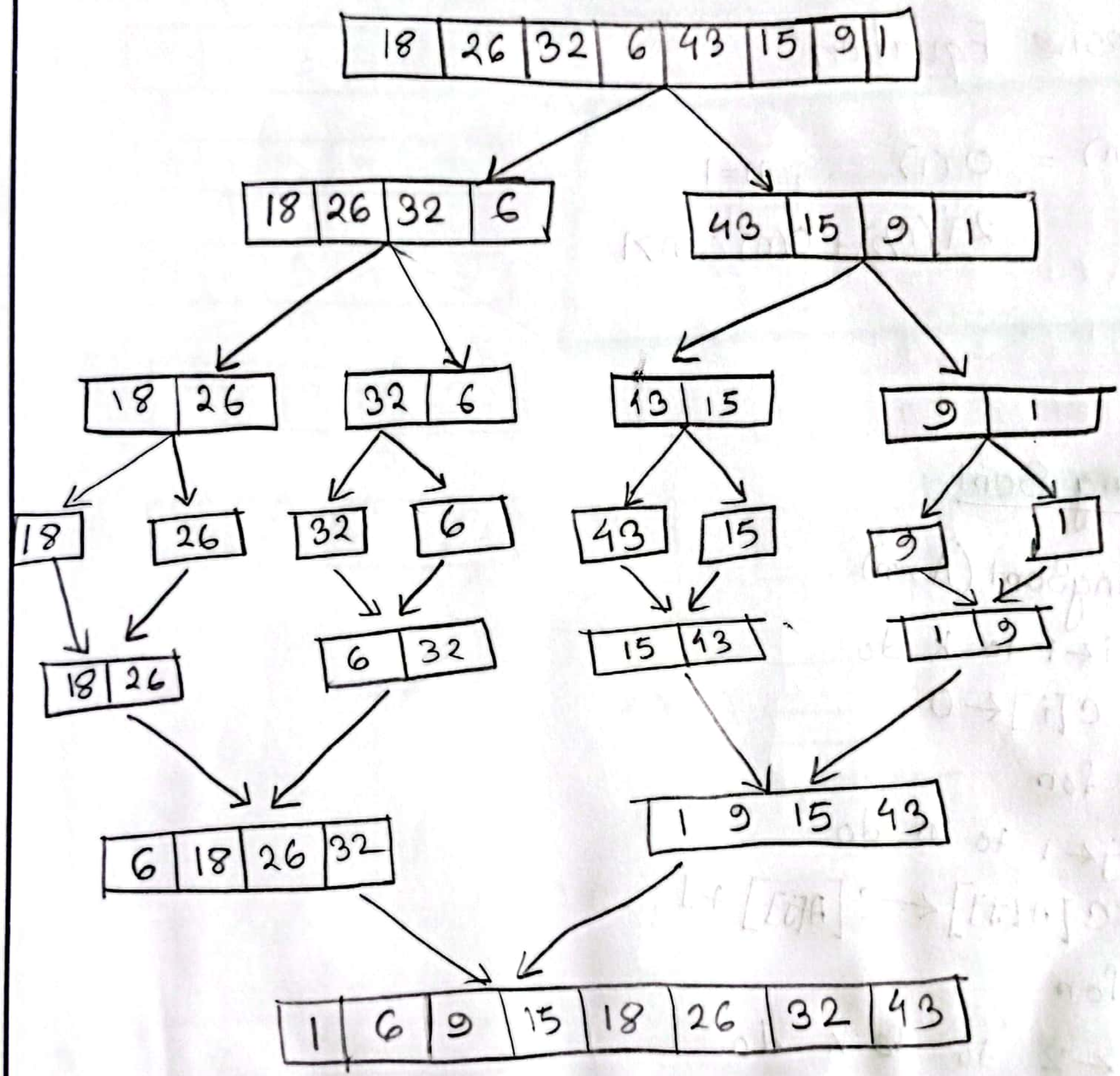
$= c\,\dfrac{n(n-1)}{2}$

$\therefore O(n^2)$

MergeSort

```
MergeSort (A, low, high)
    mid = (low + high)/2 —— 1
    if low < high then —— 1
        MageSort (A, low, mid) — n/2
        MergeSort (A, mid+1, high) — n/2
        Merge (A, low, high, mid) — n
    endif
end process

Merge (A , low, high , mid)
    i = low
    j = mid +1
    K = 0
    c = [high - low + 1]
    while i <= mid and j <= high do
        if A[i] < A[j] then
            c[k++] = A[i++]
        else
            c[k++] = A[j++]
    end while
    While (i <= mid) do
        c[k++] = A[i++]
    end while
    While j <= high do
        c[k++] = A[i++]
    end while
for (i=low, i<=high) do
A[i] = c[i-low]
end for
end process
```

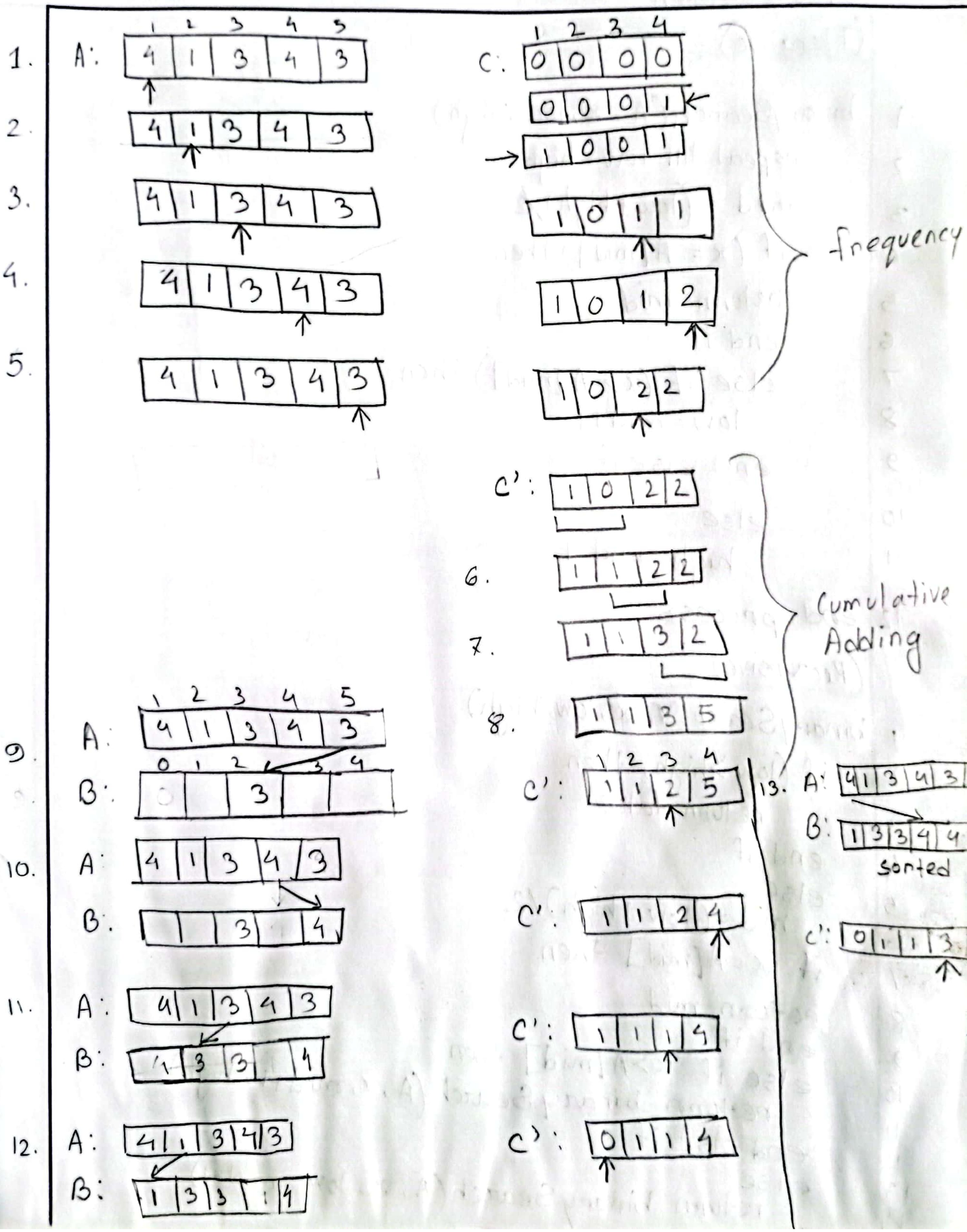

18 26 32 6 43 15 9 1
18 26 32 6
43 15 9 1
18 26
32 6
43 15
9 1
18
26
32
6
43
15
9
1
18 26
6 32
15 43
1 9
6 18 26 32
1 9 15 43
1 6 9 15 18 26 32 43

Recursive Equation:

$$T(n) = \begin{cases} O(1) & ; n=1 \\ 2T(n/2) + O(n) & ; n>1 \end{cases}$$

Counting Sort

```
countingSort(A, n)
 for i←1 to k do
       C[i]←0                          O(k)
 end for
 for j←1 to n do
       C[A[j]]←C[A[j]]+1               O(n)
 end for
 for i←2 to k do
       C[i]← C[i]+C[i-1]               O(k)
 end for
 for j←n to 1 do
       B[C[A[j]]]← A[j]                O(n)
       C[A[j]]←C[A[j]]-1
 end for
end process
                                  O(n+k)  [worst case]
```


A:
C:
frequency
C':
Cumulative Adding
B:
sorted

Binary Search

(Iterative)

```
binarySearch(A, x, low, high)
    repeat till low = high
    mid = (low + high)/2
    if (x = A[mid]) then
    return mid
    end if
    else if (x > A[mid]) then
        low = mid+1
    end else if
    else
        high = mid-1
end process
```

(Recursive)

```
binarySearch (A, x, low, high)
    if (low > high) then
        return false
    end if
    else
        mid = (low + high)/2
    if x = A[mid] then
    return mid
    end if
    else if x > A[mid] then
        return binarySearch(A, x, mid+1, high)
    end else if
    else
        return binarySearch(A, x, low, mid-1)
```

$x = 4$

3	4	5	6	7	8	9

low mid high

$4 < mid$

3	4	5	6	7	8	9

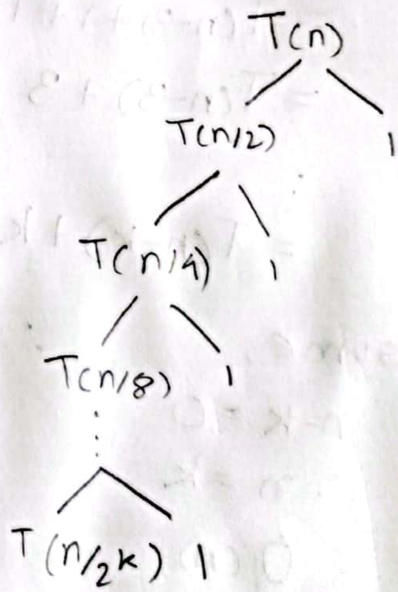
low mid high

$4 = mid$

Recursive Equation:

$$\boxed{T(n) = \begin{cases} T(n/2) + 1 & ; n > 1 \\ 1 & ; n = 1 \end{cases}}$$

$$\begin{aligned}
\therefore T(n) &= T(n/2) + 1 \\
&= [T(n/4) + 1] + 1 \\
&= [T(n/8) + 1]\ 1 + 1 \\
&= T(n/8) + 3 \\
&= T(n/2^3) + 3 \\
&\ \ \vdots \\
&= T(n/2^k) + k
\end{aligned}$$



Assume,

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

$\Rightarrow n = 2^k$

$\therefore k = \log_2 n$

$\therefore O(\log n)$

Scanned with CamScanner

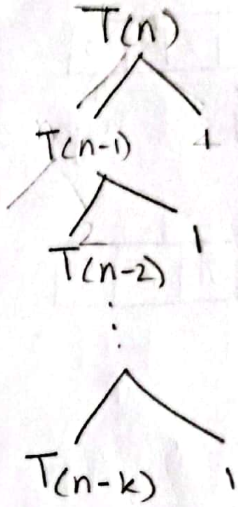
Recurrance

1.

```
void Test (int n){
    if(n>0){
        cout<<n;        ——— 1
    }
    Test(n-1);          ——— T(n-1)
}
```

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

$\therefore O(n)$



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

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

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

$= T(n-3) + 3$

$\vdots$

$= T(n-k) + k$

Assume,

$n - k = 0$

$\therefore n = k$

$\therefore T(n) = T(0) + n$

$= 1 + n$

$\therefore O(n)$

2.

```
Void Test(int n){
    if(n>0){                          ——— 1
        for(int i=0; i<n; i++){       ——— n+1
            cout<<n;                  ——— n
        }
        Test(n-1);                    ——— T(n-1)
    }
}
```

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

$= T(n-1) + n$

$\therefore T(n) = \begin{cases} 1 & ; n=0 \\ T(n-1)+n & ; n>0 \end{cases}$

3.

```
void Test(int n){
    if(n>0){
        for(i=1; i<n; i=i*2){   ]
            cout<<              ]  logn
        }                       ]
    Test(n-1)  ——————— T(n-1)
    }
```

$T(n) = T(n-1) + \log n \quad ; n>0$

$= 1 \quad ; n=0$

Scanned with CamScanner

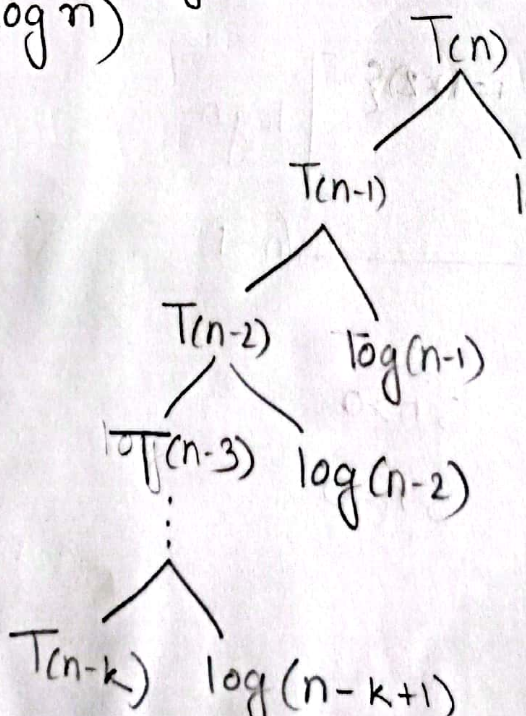
$$T(n) = T(n-1) + \log n$$
$$= T(n-2) + \log(n-1) + \log n$$
$$= T(n-3) + \log(n-2) + \log(n-1) + \log n$$
$$\vdots$$
$$= T(n-k) + \log(n-k+1) + \log(n-k+2) + \cdots + \log n$$

Assume, $n-k=0$

$\therefore n=k$

$$\therefore T(n) = T(0) + \log 1 + \log 2 + \cdots + \log n$$
$$= 1 + \log(1 \times 2 \times \cdots \times n)$$
$$= 1 + \log n!$$
$$= 1 + n\log n$$

$\therefore O(n\log n)$



4. $T(n) = \begin{cases} 1 & ; n=0 \\ 2T(n-1)+1 & ; n>0 \end{cases}$

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

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

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

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

$= 2^3 T(n-3) + 2^2 + 2^1 + 2^0$

$\vdots$

$= 2^k T(n-k) + 2^{k-1} + 2^{k-2} + \dots + 2^1 + 2^0$

Assume, $n-k=0$

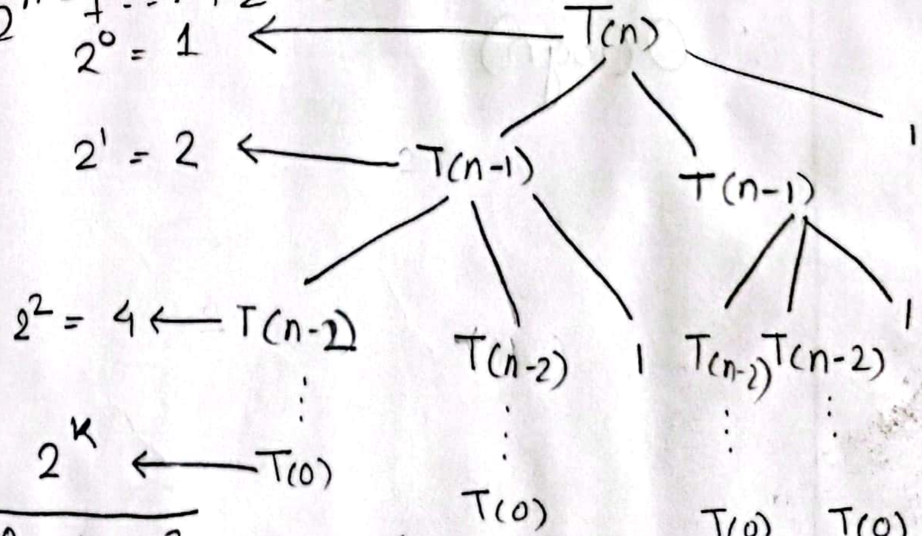
$\therefore n = k$

$\therefore T(n) = 2^n T(0) + 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0$

$= 2^n + 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0$

$= 2^{n+1} - 1$

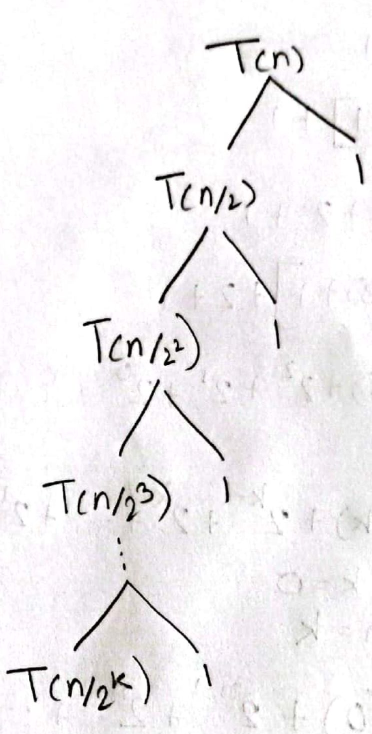
$\therefore O(2^n)$



$2^0 + 2^1 + 2^2 + \dots + 2^k$

$= 2^{k+1} - 1$

5. $T(n) = \begin{cases} 1 & ; n=1 \\ T(n/2)+1 & ; n>1 \end{cases}$



$$
\begin{aligned}
T(n) &= T(n/2) + 1 \\
&= T(n/4) + 1 + 1 \\
&= T(n/8) + 1 + 1 + 1 \\
&= T(n/2^3) + 3 \\
&\vdots \\
&= T(n/2^k) + k
\end{aligned}
$$

Assume, $2^k = \cancel{\log}\, n$

$\Rightarrow k = \log_2 n$

$\therefore T(n) = T(1) + \log_2 n$

$= 1 + \log_2 n$

$\therefore O(\log n)$

6.

$$T(n) = \begin{cases} 1 & n=1 \\ T(n/2)+n & n>1 \end{cases}$$

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

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

$$= T(n/8) + n/4 + n/2 + n = T(n/2^3) + n/2^2 + n/2^1 + n$$

$$\vdots$$

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

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

$\Rightarrow n = 2^k$

$\therefore k = \log_2 n$

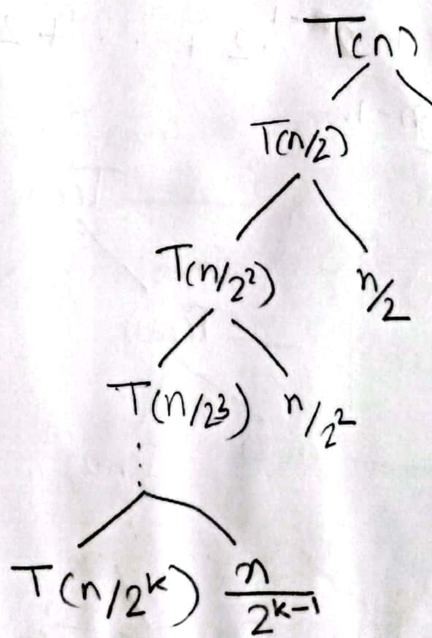
$$\therefore T(n) = T(1) + n\left(\frac{1}{2^{n-1}} + \frac{1}{2^{n-2}} + \cdots + \frac{1}{2^1} + \frac{1}{2^0}\right)$$

$$= 1 + n\,(1+1)$$

$$\cancel{= 1 + n + n}$$

$$= 1 + 2n$$

$\therefore O(n)$



7.

$$T(n) = \begin{cases} 1 & ; n = 1 \\ 2T(n-1) + n & ; n > 1 \end{cases}$$

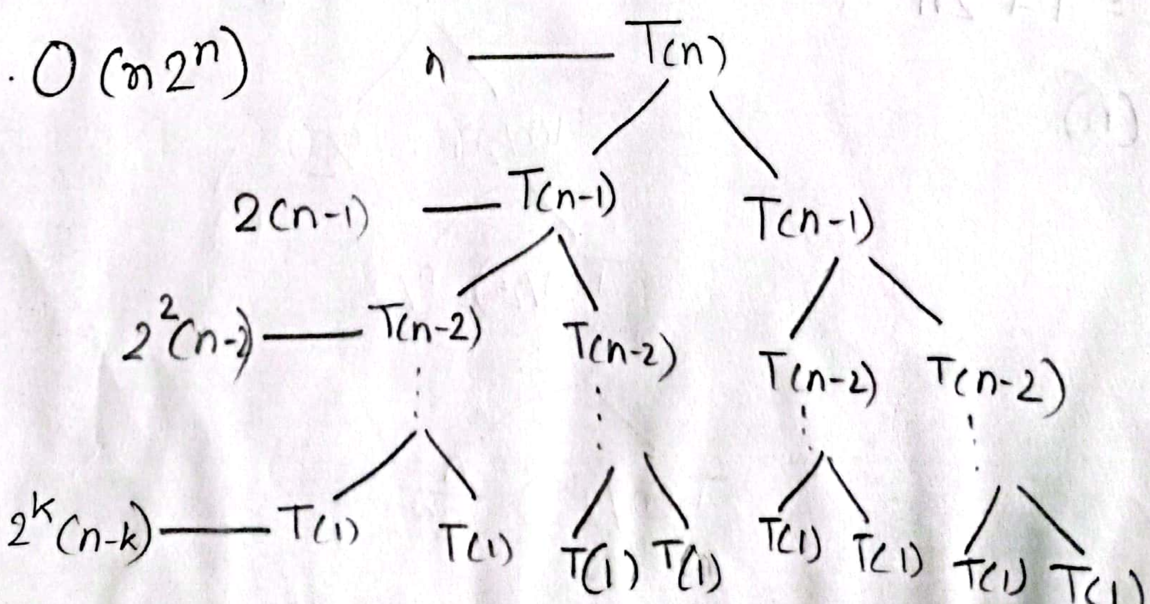
$$T(n) = 2T(n-1) + n$$
$$= 2^2 T(n-2) + 2(n-1) + n$$
$$= 2^3 T(n-3) + 2^2(n-2) + 2(n-1) + n$$
$$\vdots$$
$$= 2^k T(n-k) + 2^{k-1}(n-k+1) + \cdots + 2(n-1) + n$$

Assume, $n - k = 1$

$\therefore$ ~~n-k+~~ $k = n-1$

$$\therefore T(n) = 2^{n-1} T(1) + 2^{n-2}(2) + \cdots + 2(n-1) + n$$
$$= 2^{n-1} \cdot 1 + 2^{n-2} \cdot 2 + \cdots + 2(n-1) + 2^0 n$$
$$= 2^{n-1}\left(1 + 2^{-1} \cdot 2 + \cdots + 2^{-n}(n+1) + 2^{-n+1} n\right)$$
$$= n \cdot 2^{n-1}$$

$\therefore O(n 2^n)$



Masters Theorem

(Decreasing order)

- $T(n) = a\,T(n-b) + f(n)$

  $a > 0$, $b > 0$, $f(n) = O(n^k)$ where $k \geq 0$

  Case: 1

  $a = 1$; $O(n\,f(n))$

  Eg: (i) $T(n) = T(n-1) + 1$

  $a = 1$, $b = 1$, $f(n) = 1$

  $\therefore O(n.1) = O(n)$

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

  $a = 1$, $b = 1$, $f(n) = n$

  $\therefore O(n.n) = O(n^2)$

  (iii) $T(n) = T(n-1) + \log n$

  $a = 1$, $b = 1$, $f(n) = \log n$

  $\therefore O(n.\log n) = O(n \log n)$

Case: 2

$a > 1$; $\cancel{d}(f(n) \cdot a^{n/b})$

eg: (i) $T(n) = 2T(n-1) + 1$

$a = 2$; $b = 1$, $f(n) = 1$

$\therefore O(1 \cdot 2^{n/1}) = O(2^n)$

(ii) $T(n) = 3T(n-1) + 1$

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

$\therefore O(1 \cdot 3^{n/1}) = O(3^n)$

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

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

$\therefore O(n \cdot 2^{n/1}) = O(n 2^n)$

Case: 3

$a < 1$; $O(f(n))$

eg: $T(n) = \frac{1}{2} T(n-1) + n$

$\therefore a = \frac{1}{2}$, $b = 1$, $f(n) = n$

$\therefore O(n)$

Scanned with CamScanner

(Dividing function)

- $T(n) = aT(n/b) + f(n)$

  $a \geq 1$, $b > 1$, $f(n) = \theta(n^k \log^p n)$

Case: 1

$\log_b^a > k$, $\theta(n^{\log_b^a})$

eg: (i) $T(n) = 2T(n/2) + 1$

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

$= \theta(n^0 \log^0 n)$

$\therefore K = 0$, $p = 0$

$\therefore \log_2^2 = 1 > k$

$\therefore \theta(n^{\log_2^2}) = \theta(n^1) = \theta(n)$

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

$a = 4$, $b = 2$, $f(n) = \theta(n)$

$= \theta(n^1 \log^0 n)$

$\therefore k = 1$, $p = 0$

$\therefore \log_2^4 = 2 > k$

$\therefore \theta(n^{\log_2^4}) = \theta(n^2)$

Scanned with CamScanner

(iii) $T(n) = 8T(n/2) + n^2$

$\therefore a = 8, \quad b = 2, \quad f(n) = \theta(n^2 \cdot \log^0 n)$

$\therefore k = 2, \ p = 0$

$\therefore \log_2 8 = 3 > k$

$\therefore \theta(n^{\log_2 8}) = \theta(n^3)$

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

$a = 9, \quad b = 3, \quad f(n) = \theta(1)$

$= \theta(n^0 . \log^0 n)$

$\therefore k = 0, \ p = 0$

$\therefore \log_b a = \log_3 9 = 2 > 0$

$\therefore \theta(n^{\log_3 9}) = \theta(n^2)$

Case: 2

$\log_b a = k$, $p > -1$, $\theta(n^k \log^{p+1} n)$

$p = -1$, $\theta(n^k \log\log n)$

$p < -1$, $\theta(n^k)$

eg: (i) $T(n) = 9T(n/3) + n^2$

$a = 9, \quad b = 3, \quad f(n) = \theta(n^2)$

$= \theta(n^2 \log^0 n)$

$\therefore P = 0, \; k = 2$

$\therefore \log_3 9 = 2 = k \quad \& \quad P > -1$

$\therefore \theta(n^2 \log^1 n) = \theta(n^2 \log n)$

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

$a = 2, \; b = 2, \quad f(n) = \theta(n)$

$= \theta(n^1 \log^0 n)$

$\therefore k = 1, \; p = 0$

$\therefore \log_2 2 = 1 = k \quad \& \quad p > -1$

$\therefore \theta(n^1 \log^1 n) = \theta(n \log n)$

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

$\therefore a = 4, \; b = 2, \quad f(n) = \theta(n^2)$

$= \theta(n^2 \log^0 n)$

$\therefore k = 2, \; p = 0$

$\therefore \log_2 4 = 2 = k \quad \& \quad p > -1$

$\therefore \theta(n^2 \log n)$

(iv) $T(n) = 4T(n/2) + n^2 \log n$

$\therefore a=4, \quad b=2, \quad f(n) = \theta(n^2 \log n)$

$= \theta(n^2 \log^1 n)$

$\therefore k=2, \quad p=1$

$\therefore \log_2 4 = 2 = k \quad \& \quad p > -1$

$\therefore \theta(n^2 \log^2 n)$

(v) $T(n) = 2T(n/2) + \frac{n}{\log n}$

$\therefore a=2, \quad b=2, \quad f(n) = \theta\left(\frac{n}{\log n}\right)$

$= \theta(n^1 \log^{-1} n)$

$\therefore p=-1, \quad k=1$

$\therefore \log_2 2 = 1 = k \quad \& \quad p = -1$

$\therefore \theta(n \log\log n)$

(vi) $T(n) = 2T(n/2) + n \log^{-2} n$

$\therefore a=2 \quad b=2 \quad f(n) = \theta(n^1 \log^{-2} n)$

$\therefore p=-2 \quad k=1$

$\therefore \log_a b = \log_2 2 = 1 = k \;\&\; p < -1$

$\therefore \theta = (n)$

Case: 3

$\log_b^a < k$, $p \geq 0$, $\theta(n^k \log^p n)$

$p < 0$, $\theta(n^k)$

eg: (i) $T(n) = T(n/2) + n^2$

$a = 1$, $b = 2$, $f(n) = \theta(n^2 \log^0 n)$

$\therefore k = 2$, $p = 0$

$\therefore \log_b^a = \log_2^1 < k$ & $p = 0$

$\therefore \theta(n^2 \log^0 n) = \theta(n^2)$

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

$a = 2$, $b = 2$, $f(n) = \theta(n^2 \log^0 n)$

$\therefore k = 2$, $p = 0$

$\therefore \log_2^2 = 1 < k$ & $p = 0$

$\therefore \theta(n^2 \log^0 n) = \theta(n^2)$

(iii) $T(n) = T(n/2) + \frac{n^3}{\log n}$

$a = 1$, $b = 2$, $f(n) = \theta(n^3 \log^{-1} n)$

$\therefore k = 3$, $p = -1$

$\therefore \log_2^1 < k$ & $p < 0$

$\therefore \theta(n^3)$

Greedy method

General

```
Greedy (a,n){
   for i<-1 to n do
      x = select(a)
      if Feasible(x) then
         solution = solution + x
      end if
   end for
}
```

n=5

a	a_0	a_1	a_2	a_3	a_4

Knapsac Problem

n=7

m=15 kg

Obj	0	1	2	3	4	5	6	7
profit	p	10	5	15	7	6	18	3
weight	w	2	3	5	7	1	4	1
	$\frac{P}{w}$	5	1.66	3	1	6	4.5	3

$0 \le x \le 1$

$x \left(\begin{matrix} 1 & 2/3 & 1 & 0 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \right)$

Verification:

$\Sigma x_i w_i = 1 \times 2 + \frac{2}{3} \times 3 + 1 \times 5 + 0 \times 7 + 1 \times 1 + 1 \times 4 + 1 \times 1$

$= 15$ kg

15-1 = 14

14-2 = 12

12-4 = 8

8-5 = 3

3-1 = 2

2-2 = 0 kg

$\Sigma x_i p_i = 1 \times 10 + \frac{2}{3} \times 5 + 1 \times 15 + 0 \times 7 + 1 \times 6 + 1 \times 18 + 1 \times 3$

$= 54.6$

Constraint : $\Sigma x_i w_i \le m$

Object : $\Sigma x_i p_i$ max

$W = 16\,kg$

item	weight	value	V/W	chosen
i_1	6	6	1	1
i_2	10	2	0.2	0
i_3	3	1	0.33	$1/3$
i_4	5	8	1.6	1
i_5	1	3	3	1
i_6	3	5	1.66	1

$16-1 = 15$

$15-3=12$

$12-5=7$

$7-6=1$

$1-1=0$

$$\sum i_i W_i = 1\times6 + 0\times10 + \tfrac{1}{3}\times3 + 1\times5 + 1\times1 + 1\times3$$

$$= 6 + 0 + 1 + 5 + 1 + 3$$

$$= 16$$

$$\sum i_i V_i = 1\times6 + 0\times2 + \tfrac{1}{3}\times1 + 1\times8 + 1\times3 + 1\times5$$

$$= 6 + 0 + \frac{1}{3} + 8 + 3 + 5$$

$$= 22.33$$