1.a.

$$t_n = \begin{cases} 1 & , \text{if } n=0 \\ 4t_{n-1} - 2^n & , \text{otherwise} \end{cases}$$

$\Rightarrow \quad t_n = 4t_{n-1} - 2^n$

$\Rightarrow \quad t_n - 4t_{n-1} = -2^n$

$\Rightarrow \quad -t_n + 4t_{n-1} = 2^n$

$$b^n p(n)$$
$$\downarrow \quad \downarrow$$
$$2^n \quad n^0$$

$\therefore \; b = 2$

$d = 0$

$\Rightarrow \quad -t_n + 4t_{n-1} = 2^n$

$(-x+4)\;(x-2) = 0$

$\therefore \; \cancel{x} \; r_1 = 4, \; r_2 = 2$

$-x^n + 4x^{n-1} = 0$

$-x + 4 = 0$

$(x-2)^{0+1}$

$(x-2)$

$\therefore \; t_n = c_1 4^n + c_2 2^n \longleftarrow$ 1

for $n=0$

$t_0 = c_1 4^0 + c_2 2^0$

$1 = c_1 + c_2 \longleftarrow$ 2

for $n=1$

$t_1 = c_1 4^1 + c_2 2^1$

$t_1 = 4c_1 + 2c_2$

Page No.

Date :

For t_1

$$t_n = 4t_{n-1} - 2^n$$

$$t_1 = 4t_{1-1} - 2^1$$

$$= 4t_0 - 2$$

$$= 4 \times 1 - 2$$

$$= 2$$

$\therefore$ $t_1 = 4C_1 + 2C_2$

$2 = 4C_1 + 2C_2 \leftarrow (3)$

(2) $\times 2 -$ (3) $\Rightarrow$

$$\cancel{2} = 2C_1 + \cancel{2C_2}$$

$$-\cancel{2} = -4C_1 \cancel{\mp 2C_2}$$

$$0 = -2C_1$$

$$\boxed{C_1 = 0}$$

$\cancel{4 = 4C_1 + 4C_2}$

$2 = 4C_1 + 2C_2$

$2 = 2C_2$

$\cancel{C_2 = 1}$

$\therefore$ $C_1 + C_2 = \phi$

$$\boxed{C_2 = 1}$$

(1) $\Rightarrow$

$\therefore$ $t_n = (0) 4^n + (1) 2^n$

$\therefore$ Time complexity

Page No.
Date :

~~3.b~~

2.a. Theory

2.b.
```
A()
{   int i,j,k
    for (i=n/2; i<=n, i++)
      for (j=1; j<=n; j=2*j)
        for (k=1; k<=n; k=2*k)
          printf ("PCE Nagpur")
}
```

$$\Rightarrow 1 + \sum_{i=n/2}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (1)$$

$$\Rightarrow 1 + \sum_{i=n/2}^{n} \sum_{j=1}^{n} (1\times n)$$

$$\Rightarrow 1 + \sum_{i=n/2}^{n} \sum_{j=1}^{n} (n)$$

$$\Rightarrow 1 + n \sum_{i=n/2}^{n} \sum_{j=1}^{n} (1)$$

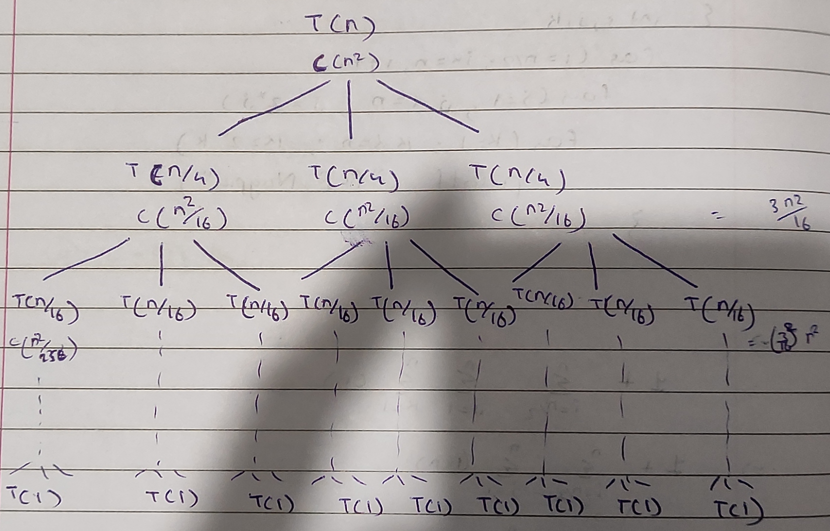
$$\Rightarrow 1 + n \sum_{i=n/2}^{n} (n)$$

$$\Rightarrow 1 + n^2 \sum_{i=n/2}^{n} (1)$$

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

$\therefore$ Time complexity $= \Theta(n^3)$

3.a. $T(n) = 3T(n/4) + \Theta(n^2)$



- Subproblem at depth i

$\cancel{y = \frac{n}{2^i}}$ $\quad y = \frac{n}{4^i}$

$4^i = n$

$i \lg 4 = \lg n$

$\boxed{i = \lg_4 n}$

Page No.
Date :

No. of nodes at particular depth $= 3^i$

Cost of node at particular depth $= \frac{C(n^2) \times 3^i}{3^i}$

$= C(n^2)$

Cost of Recursion Tree $T(n)$

$= C(n^2) + C\frac{3}{16}n^2 + C \cdot \frac{9}{16}n^2 + \cdots$

$= Cn^2 \left[1 + \frac{3}{16} + \frac{9}{16} + \cdots \right]$

$= Cn^2 \left[\frac{1}{1 - 3/16} \right]$

$1 + r + r^2 + \cdots \quad r<1$

$= \frac{1}{1-r}$

$= Cn^2 \left[\frac{1}{\frac{13}{16}} \right]$

$= Cn^2 \left[\frac{16}{13} \right]$

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

b.

i. $T(n) = 6T(n/3) + n^2 \log n$

$\Rightarrow \quad T(n) = 6T(n/3) + n^2 \log n$

$a = 6, \quad b = 3$

Page No.
Date :

4.a. $T(n) = 2T(\sqrt{n}) + \log n$

$\Rightarrow$ put $m = \log_2 n$ in eqn

$\therefore \quad n = 2^m$

i.e. $\quad n^{1/2} = 2^{m/2}$

$\Rightarrow \sqrt{n} = 2^{m/2}$

$\therefore \quad T(2^m) = 2T(2^{m/2}) + \log(2^m)$

$\Rightarrow \quad T(2^m) = 2T(2^{m/2}) + m \log 2$

$\Rightarrow \quad T(2^m) = 2T(2^{m/2}) + m \quad \leftarrow (2)$

Substitute $\quad S(m) = T(2^m)$

$\therefore \quad S(m) = 2S(m/2) + m \quad \leftarrow (3)$

This is logarithmic recurrence

$\therefore$ By Master Theorem

$\therefore \quad a = 2 \quad b = 2 \quad , f(n) = m$

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

$\therefore \quad m^{\log_b a} = f(n)$

$\therefore$ case (2)

$\therefore \quad T(m) = \Theta(n^{\log_b a} \lg n)$

$= \Theta(m \lg m)$

Page No.
Date :

4.b. Theory.

$T(n) = aT(n/b) + \Theta(n^k \log^p n)$

~~Set.~~

3.b.

1. $T(n) = 6T(n/3) + n^2 \log n$

$\Rightarrow$ $a = 6$, $b = 3$, $f(n) = n^2 \log n$, $k = 2$, $P = 1$

$b^k = 3^2 = 9$

$\therefore a < b^k$

$\therefore$ Case 3

$\because P \geq 0$

$\therefore T(n) = \Theta(n^k \log^p n)$

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

2. $T(n) = \sqrt{2}\, T(n/2) + \log n$

$\Rightarrow$ $a = \sqrt{2}$, $b = 2$, $f(n) = \log n$, $K = 0$, $P = 1$

$b^k = 2^0 = 1$

$\therefore a > b^k$

$\therefore$ Case 1

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

$= \Theta(n \log \sqrt{2})$

5. a.

Algorithm:

```
Binary Search (a,i,l, x)
{
    if (l = i) then            // single element in Array.
    {
        if (x = a(i)) then return i;
        else return 0;
    }
    else    // more than one ele in array
    {
        mid = (i+1)/2        // divide array from mid
        if (x = a(mid)) then return mid;
        else if (x < a(mid)) // Compare with mid ele
          then
// Go to left side →     return Binary search (a, i, mid-1, x)
        else
// Go to right side →    return Binary Search (a, mid+1, l, x);
    }
```

→ Stepwise execution of search element X = 45

A = <9, 12, 15, 24, 30, 36, 45, 70>

⇒

Step 1: We want to search 45, therefore X = 45

Page No.
Date :

Given array is

a ⇒	9	12	15	24	30	36	45	70
index ⇒	1	2	3	4	5	6	7	8
				mid				

Binary Search $(a, 1, 8, 45)$

check for Single or multi element : if$(l = l)$ ← False

∴ Array contain multi element.

compute mid $= (i+l)/2 = (1+8)/2 = 9/2 = 4.5$
$= 4$

check with mid $x = a(mid)$
$45 = a(mid)$ ← False

Check left or right : $x > a(mid)$ ← True

∴ Go to right : Binary Search $(a, mid+1, l, x)$
Binary Search$(a, 5, 8, 45)$

Step 2 :

Now we have.

a	30	36	45	70
index	5	6	7	8
	i	mid		l

Binary Search $(a, 5, 8, 45)$

Check for single or multi ele : if$(l = i)$ ~~True~~
← False.

Page No.
Date :

∴ Array contain multi ele.

compute mid = $(i+l)/2$ = $(5+8)/2$ = $13/2 = 6.5$
$= 6$

check with mid: $x = a(mid)$
$45 = a(mid)$ ← false

Check for left or right : $x > a(mid)$

∴ Go to right :

Binary Search (a, mid+1, l, x)
Binary Search (a, 7, 8, 45)

Step3: Now we have:

a	45	70
index	7	8

mid

Binary search (a, 7, 8, 45)

Check for single or multi element : $(l = i)$ ← false

∴ Array contain multi ele.

Page No.
Date :

Compute mid = (i+l)/2 = (7+8)/2 = (15)/2 = 7.5
=7

Check with mid ⇒ x = a(mid)
45 = a(mid) ⟵ True.

∴ Search is esuccessful
Element x = 45 found at i = 7

→ Complete Scenaero.

Phase 1 Divide.

a	9	12	15	24	30	36	45	70
index	1	2	3	4	5	6	7	8
	i			mid				l

Binary Search (a, 1, 8, 45)

a	30	36	45	70
index	5	6	7	8
	i	mid		l

Binary Search (a, 5, 8, 45)

a	45	70
index	7	8
	i mid	l

Binary Search (a, 7, 8, 45)
Search succesful
element x = 24 found at i=5

Page No.
Date :

Phase 2 conquer

Search is succesful

element $x=24$ — Binary Search(a, 1, 8, 45)
found at $i=5$

Binary Search(a, 5, 8, 45)

Binary Search (a, 7, 8, 45)

→ Partition (A,

$x \leftarrow$
$i \leftarrow$
for
i

exch
ret

Level 1 (Divide.) Pha

Page No.
Date :

7.b

Algorithm:

```
Merge (A,p,q,r)          //compute size of left subarray
  n1 <- q-p+1            // ----"---- right --"--
  n2 <- r-q
  Create array L(1,2,----n1+1) and R(1,2,,---n2+1)
  for i <- 1 to n1
     do L(i) <- A(p+i-1)  // copy data int left subarray
  for j <- 1 to n2
     do R(j) <- A(q+j)    //copy data int right subarray
  L (n1+1) <-             //set sentinel value
  R (n2 +1) <-            // --"------
  i <- 1
  j <- 1
  for k <- p to r         // Merge left subarray and right in ascending
     do if L(i) <= R (j)
        then A(k) <- L(i)
             i <- i+1
        else
             A(k) <- R(j)
             j <- j+1
```

→ $A = \langle 15, 10, 5, 20, 25, 30, 40, 35 \rangle$

Phase 1. Divide at level 1.

A	15	10	5	20	25	30	40	35
index	1	2	3	4	5	6	7	8
	p							r

mergesort (A,p,r)

mergesort (A,1,8)

Page No.
Date :

~~then~~ if (p < r) then
q ← (p+r)/2
mergesort (A,p,q)
~~i~~ merge sort (A, q+1, r)

if (1<8) --- True
q=(1+8)/2=4.5=4
merge sort(A,1,4)
merge sort (A,5,8)

A	15	10	5	20
index	1	2	3	4
	p			q

A	25	30	40	35
index	5	6	7	8
	q+1			r

Divide at level 2

A	15	10	5	20
index	1	2	3	4

merge sort (A,1,4)

A	25	30	40	35
index	5	6	7	8

merge sort (A,5,8)

Phase

if (1<4)---- True
q=(1+4)/2=2.5=2
merge sort (A,1,2)
merge sort (A,3,4)

if (1<8)--- True
q=(5+8)/2=6.5=6
merge sort (A,5,6)
merge sort (A,7,8)

A	15	10
index	1	2
	p	q

A	5	20
index	3	4
	q+1	r

A	25	30
index	5	6
	p	q

A	40	35
index	7	8
	q+1	r

Divide at level 3

A	15	10
index	1	2
	p	q

A	5	20
index	3	4
	q+1	r

merge sort (A,1,2)
if (1<2) --- True
q=(1+2)/2=1.5=1

merge sort (A,3,4)
if(3<4) --- True
q=(3+4)/2=3.5=3

Page No.
Date :

merge sort (A,1,1)
merge sort (A,2,2)

A	15	A	10
index	1	index	2
	p,q		q+1,r

merge sort (A,3,3)
merge sort (A,4,4)

A	5	A	20
index	3	index	4
	p,q		q+1,r

Similarly

A	25	A	30	A	40	A	35
index	5	index	6	index	7	index	8
	p,q		q+1,r		p,q		q+1,r

Phase 2 Conquers.

A	15	A	10
index	1	index	2

merge (A,p,q,r)
merge (A,1,1,2)

A	5	A	20
index	3	index	4

merge (A,p,q,r)
merge (A,3,3,4)

A	25	A	30
index	5	index	6

merge(A,p,q,r)
merge (A,5,5,6)

A	40	A	35
index	7	index	8

merge(A,p,q,r)
merge(A,7,7,8)

A	10	15	A	5	20
index	1	2	index	3	4

merge(A,1,2,4)

A	25	30
index	5	6

A	35	40
index	7	8

merge (A,5,6,8)

A	5	10	15	20
index	1	2	3	4

A	25	30	35	40
index	5	6	7	8

merge (A, 1, 4, 8)

A	5	10	15	20	25	30	35	40
index	1	2	3	4	5	6	7	8

1.b , 6.a, 6.b , 7.a , 9.a, 9.b, 10.a, 10.b x

2.a
4.b } Theory
8.a

8.b → Sus

Binary Search

Binary Search (a, 7, 8,

5.b

Algo:

Quick Sort (A, P, r)
{ if (P < r) then
q ← Partition (A, P, r)
Quick sort (A, P, q-1)
Quick sort (A, q+1, r)
}

→ A = {13, 19, 7, 5, 12, 9}

⇒ ~~Partition (A, P, r)~~
~~x ← A(r)~~
~~i ← P-1~~
~~for j ← P to r-1~~

→ Partition (A, p, r)

```
x ← A(r)              // set last ele as pivot
i ← p-1
for j ← p to r-1      // from 1st to last ele
    if A(j) <= x      // compare current ele with pivot
        i ← i+1       // if current is smaller than pivot
        exchange (A(i), A(j))   // exchange
exchange ( A(i+1), A(r))   // exchange current with pivot
return i+1            // Return index at which pivot is stored
```

Level 1 (Divide.) phase 1.

→						
A	13	19	7	5	12	9
index	1	2	3	4	5	6
	p					r

Quick Sort (A, 1, 6)
↓

if (1<6) - True

q ← partition (A, 1, 6)

q = 3

quick sort (A, 1, 2)

Quick sort (A, 4, 6)

Step 1: Set pointers and pivot

A	13	19	7	5	12	9	Pivot 9
index	1	2	3	4	5	6	r
	p					r	

Page No.
Date :

Partition (A, l, b)

$\downarrow$

$x \leftarrow A(r)$

$x \leftarrow A(6) \leftarrow 9$

$x = 9$

$i \leftarrow p-1$

$i \leftarrow 1-1$

$i = 0$

Step 2 iterations.

A	13	19	7	5	12	9		9
index 0	1*	2	3	4	5	6		x
i	p					r		Pivot

for j = 1

if A(j) <= x i.e. 13 <= 9 False

No need to increment i

No need exchange (A(i), A(j))

A	13	19	7	5	12	9		9
index	1	2*	3	4	5	6		x
0	p					r		Pivot

for j = 2

if A(j) <= x i.e 19 <= 9 False

No need to increment i

No need exchange (A(i), A(j))

A		13	19	7*	5	12	9	
index	0	1	2	3	4	5	6	
	•	p					j	

for j = 3

if $A(j) <= x$ ie $7 <= 9$ true

$i \leftarrow i+1$ i.e. $i = 0+1 = 1$

exchange $(A(i), A(j))$ i.e. exchange $(A(1), A(3))$

$A(1) = 7$

$A(3) = 13$

A		7	19	13	5*	12	9	9
index	0	1	2	3	4	5	6	x
		p •					j	Pivot

for j = 4

if $A(j) <= x$ i.e. $5 <= 9$ True.

$i \leftarrow i+1$ i.e. $i = 1+1 = 2$

exchange $(A(i), A(j))$ i.e. Exchange $(A(2), A(4))$

$A(2) = 5$

$A(4) = 13$

A		7	5	13	19	12*	9	9
index	0	1	2	3	4	5	6	x
		p	•				j	Pivot

for j = 5

if $A(j) <= x$ i.e. $12 <= 9$ false

No need to increment i

No need to exchange $(A(i), A(j))$

A		7	5	13	19	12	9*
index	0	1	2	3	4	5	6
		p	•				j

Page No.
Date :

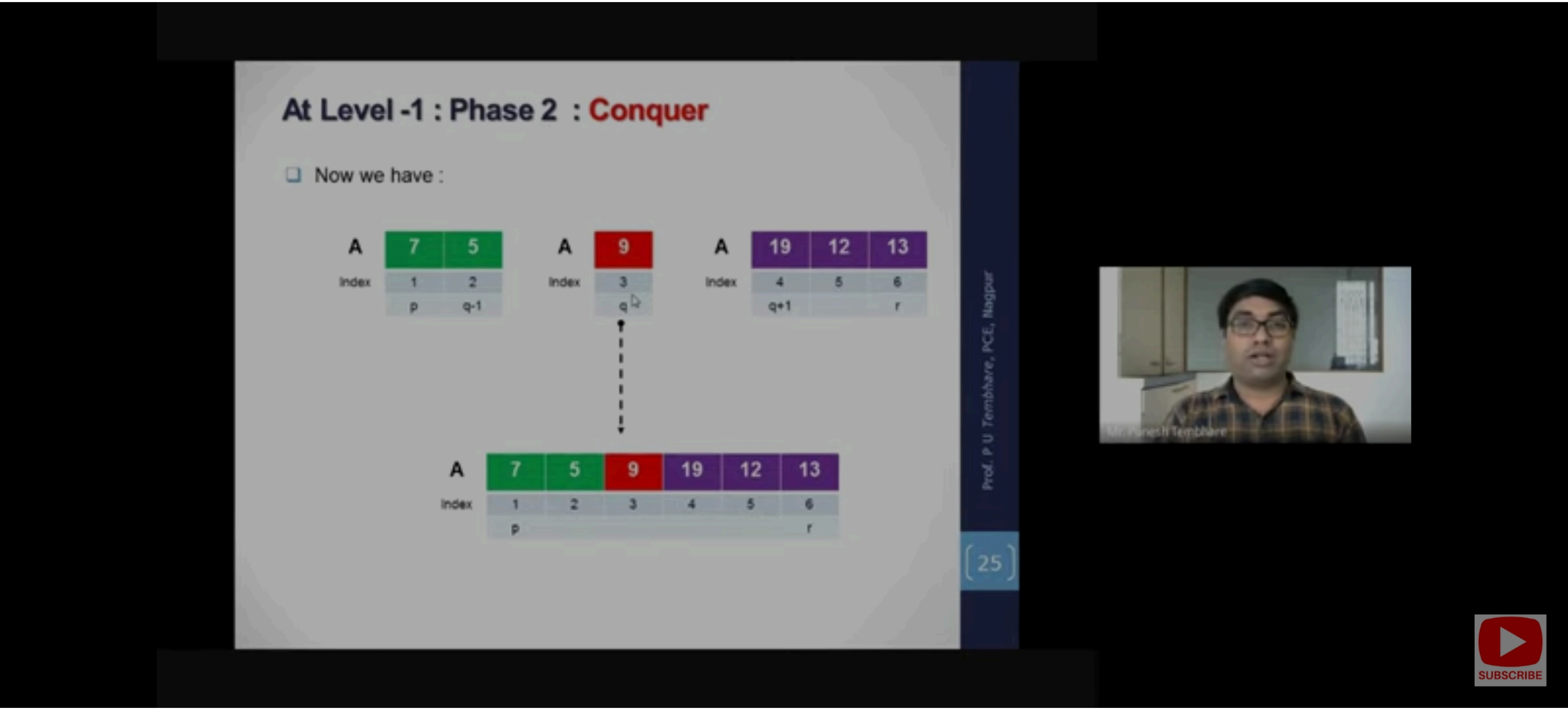
Step 3 Exchang pivot with A(i+1)

exchange (A(i+1), A(r))
i.e. exchange (A(2+1), A(6))
exchange (A(3), A(6))
A(3) = 9
A(6) = 13
return i+1 = 3

A	7	5	9	19	12	13	9
index	1	2	3	4	5	6	x
0	p					r	Pivot

Algorithm:

```
Binary Search ( a, i, l, x )
{
   if ( l = i) then
        {
          if ( x = a(i) ) then return i ;
          else return 0 ;
        }
   else
   {
        mid = ( i + l ) / 2 ;
        If (x = a(mid) ) then return mid;
        else if ( x < a(mid) )
             then
                return Binary Search (a, i, mid-1, x)
             else
                return Binary Search (a, mid+1 ,l, x);
   }
```



At Level -1 : Phase 2 : Conquer
Now we have :
A 7 5
Index 1 2
p q-1
A 9
Index 3
q
A 19 12 13
Index 4 5 6
q+1 r
A 7 5 9 19 12 13
Index 1 2 3 4 5 6
p r
Prof. P U Tembhare, PCE, Nagpur
25
SUBSCRIBE

At Level - 2 : Phase 1 : Divide

- Now we have :

A	7	5
Index	1	2
	p	q-1

A	9
Index	3
	q

A	19	12	13
Index	4	5	6
	q+1		r

Quick Sort (A,1,2)

if (1 < 2) --- TRUE
q ← Partition (A, 1, 2)
q = 1

Quick Sort (A,4,6)

if (4 < 6) --- TRUE
q ← Partition (A, 4, 6)
q = 5

A	5
Index	1
	p

A	7
Index	2
	q

A	12
Index	4
	p

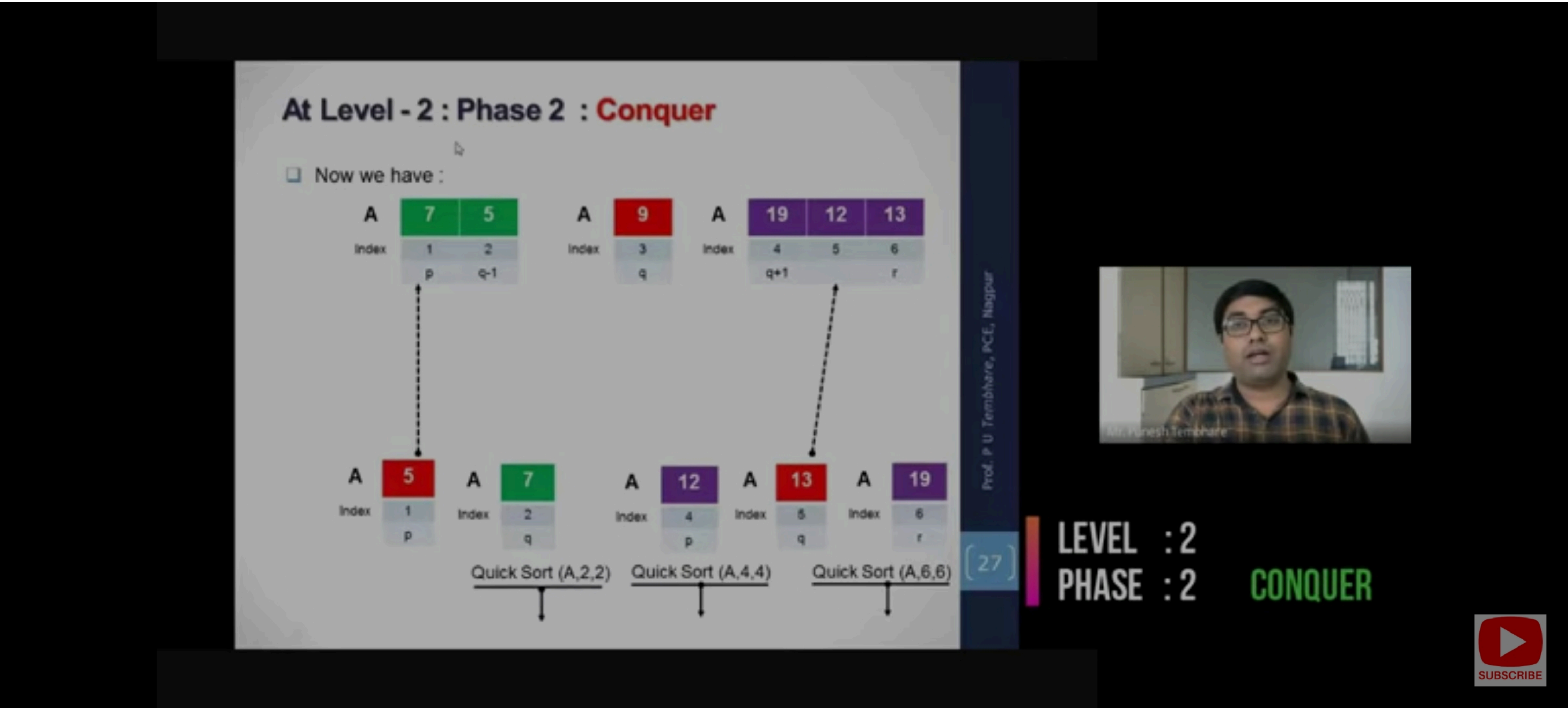
A	13
Index	5
	q

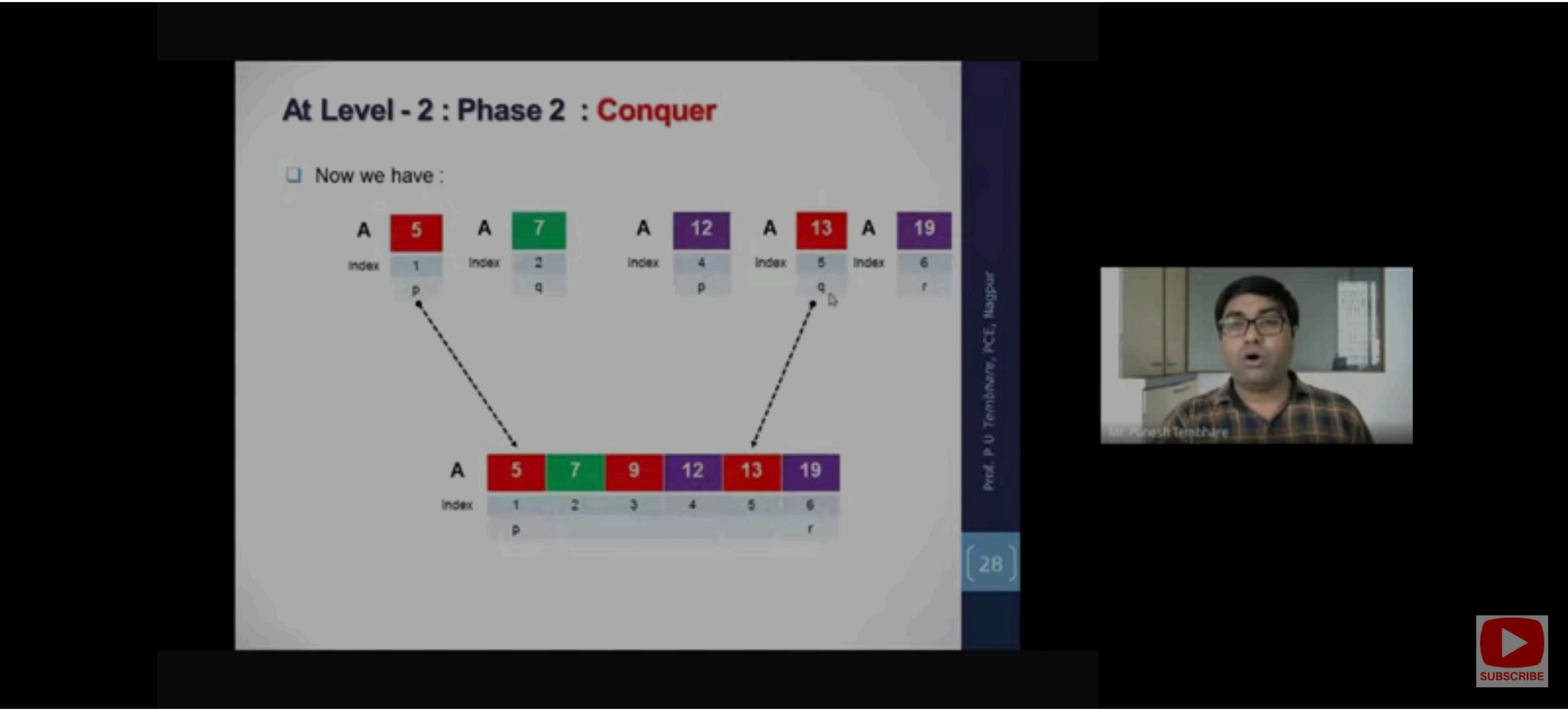
A	19
Index	6
	r

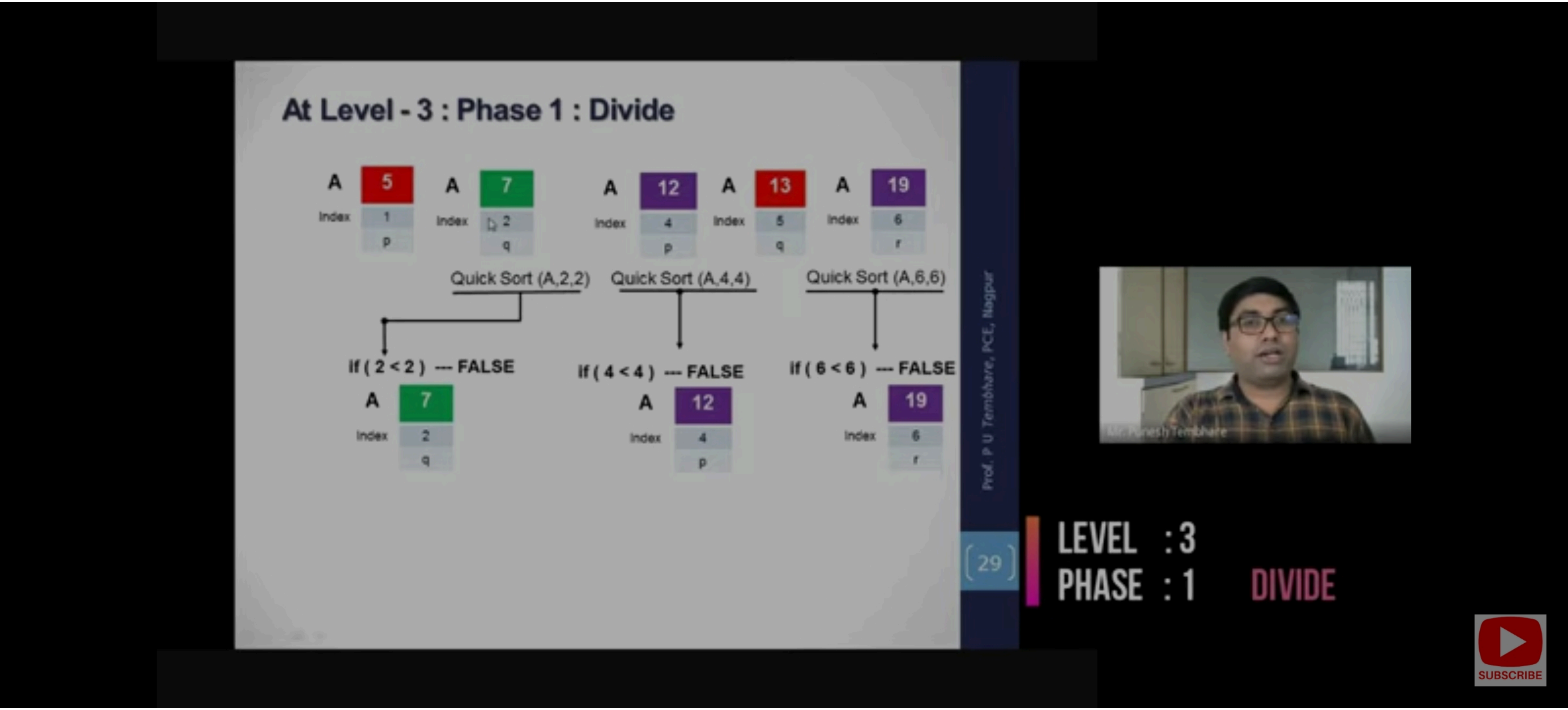
Quick Sort (A,2,2) Quick Sort (A,4,4) Quick Sort (A,6,6)

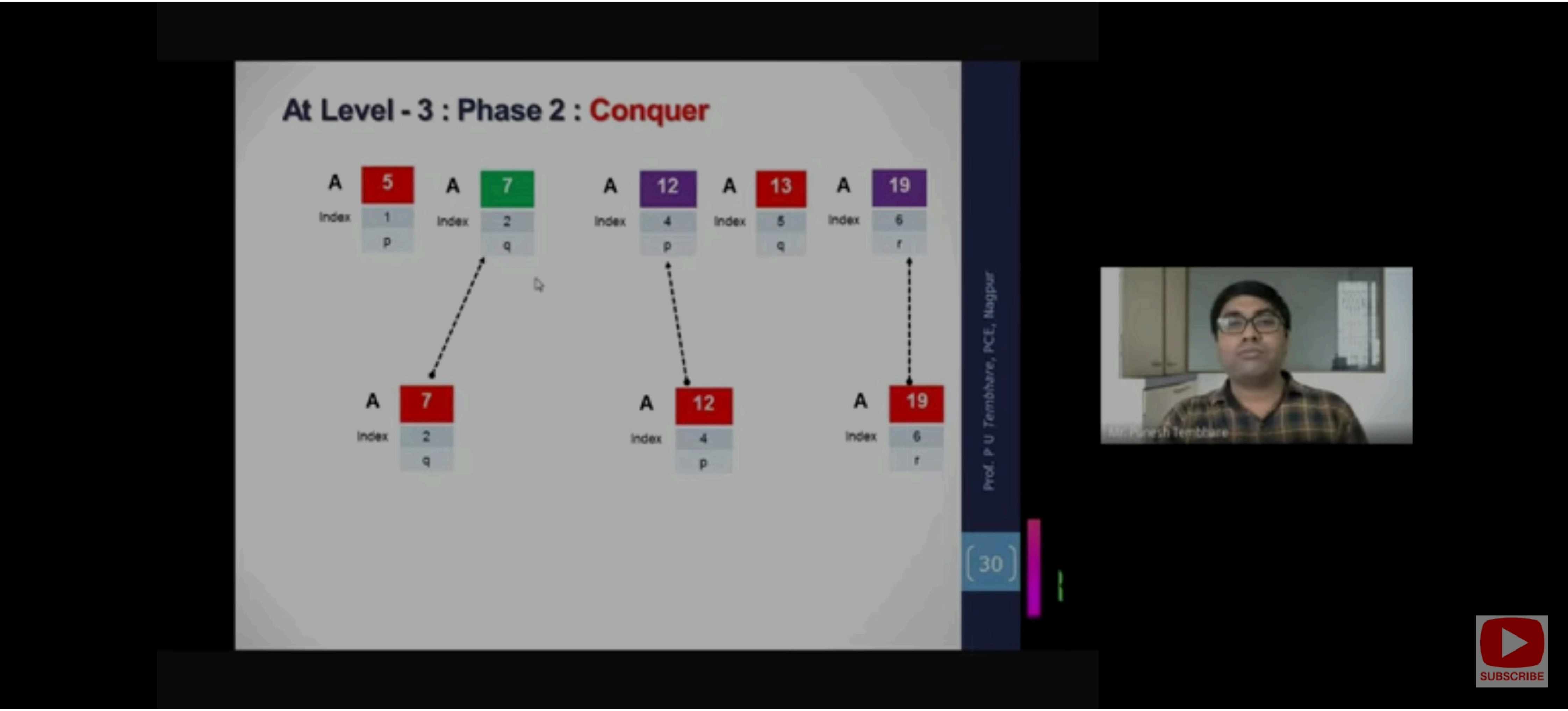
Prof. P U Tembhare, PCE, Nagpur

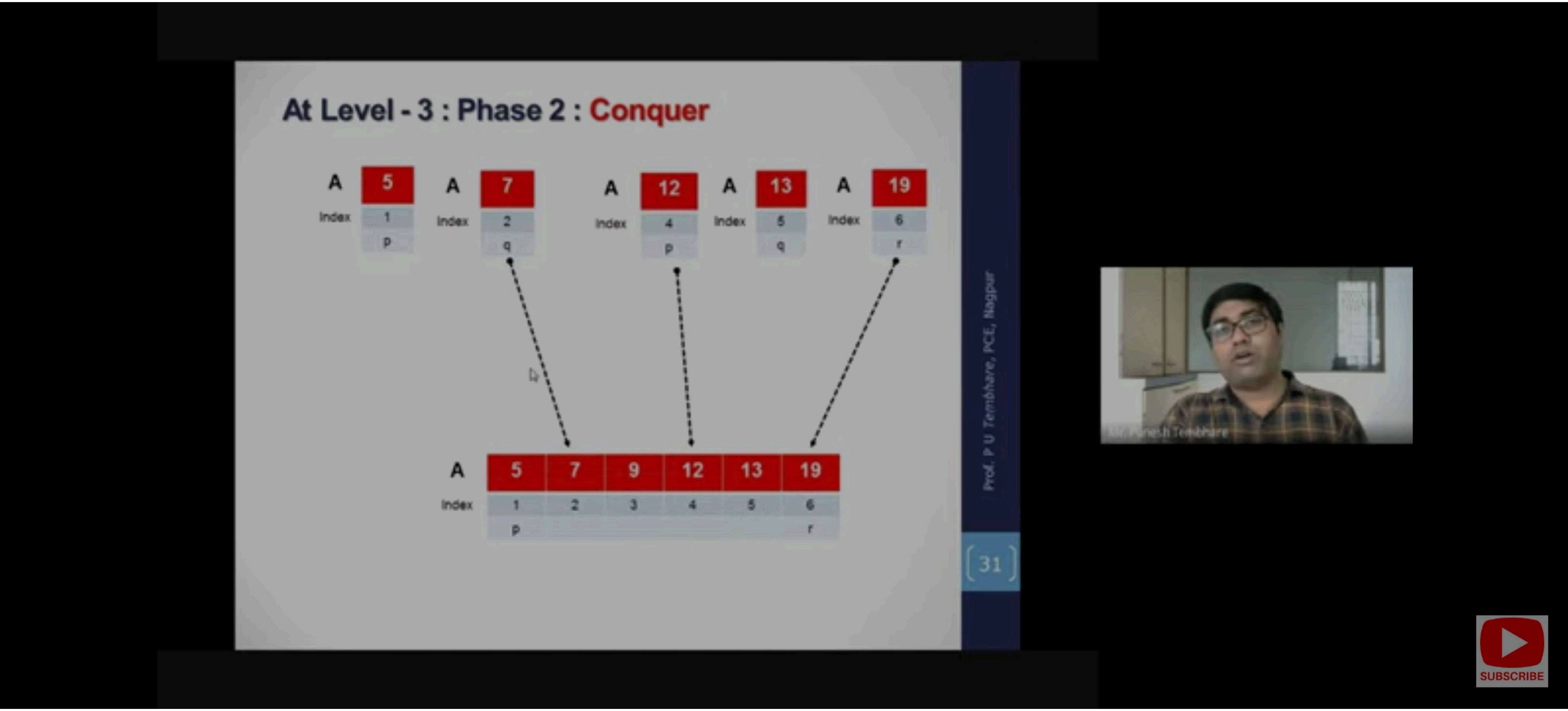
LEVEL : 2
PHASE : 1 DIVIDE


At Level - 2 : Phase 2 : Conquer
Now we have :
A 7 5
Index 1 2
p q-1
A 9
Index 3
q
A 19 12 13
Index 4 5 6
q+1 r
A 5
Index 1
p
A 7
Index 2
q
A 12
Index 4
p
A 13
Index 5
q
A 19
Index 6
r
Quick Sort (A,2,2)
Quick Sort (A,4,4)
Quick Sort (A,6,6)
Prof. P U Tembhare, PCE, Nagpur
27
LEVEL : 2
PHASE : 2
CONQUER
SUBSCRIBE


At Level - 2 : Phase 2 : Conquer
Now we have :
A 5 Index 1 p
A 7 Index 2 q
A 12 Index 4 p
A 13 Index 5 q
A 19 Index 6 r
A 5 7 9 12 13 19
Index 1 2 3 4 5 6
p r
Prof. P U Tembhare, PCE, Nagpur
28
SUBSCRIBE


At Level - 3 : Phase 1 : Divide
A 5 Index 1 p
A 7 Index 2 q
A 12 Index 4 p
A 13 Index 5 q
A 19 Index 6 r
Quick Sort (A,2,2)
Quick Sort (A,4,4)
Quick Sort (A,6,6)
If (2 < 2) --- FALSE
A 7 Index 2 q
If (4 < 4) --- FALSE
A 12 Index 4 p
If (6 < 6) --- FALSE
A 19 Index 6 r
Prof. P U Tembhare, PCE, Nagpur
29
LEVEL : 3
PHASE : 1
DIVIDE
SUBSCRIBE


At Level - 3 : Phase 2 : Conquer
A 5 Index 1 p
A 7 Index 2 q
A 12 Index 4 p
A 13 Index 5 q
A 19 Index 6 r
A 7 Index 2 q
A 12 Index 4 p
A 19 Index 6 r
Prof. P U Tembhare, PCE, Nagpur
30
SUBSCRIBE


At Level - 3 : Phase 2 : Conquer
A
5
Index
1
p
A
7
Index
2
q
A
12
Index
4
p
A
13
Index
5
q
A
19
Index
6
r
A
5 7 9 12 13 19
Index
1 2 3 4 5 6
p
r
Prof. P U Tembhare, PCE, Nagpur
31
SUBSCRIBE