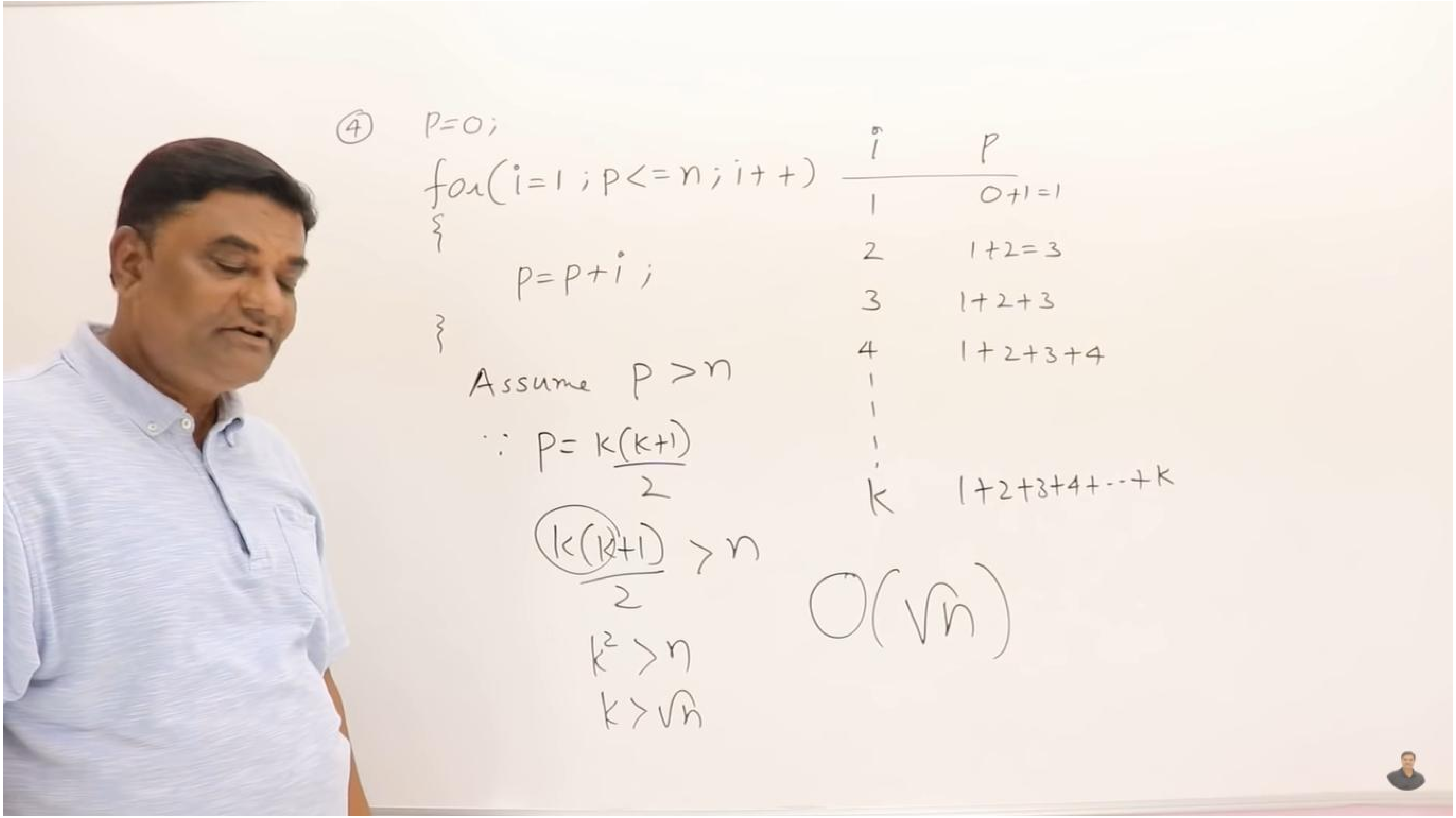
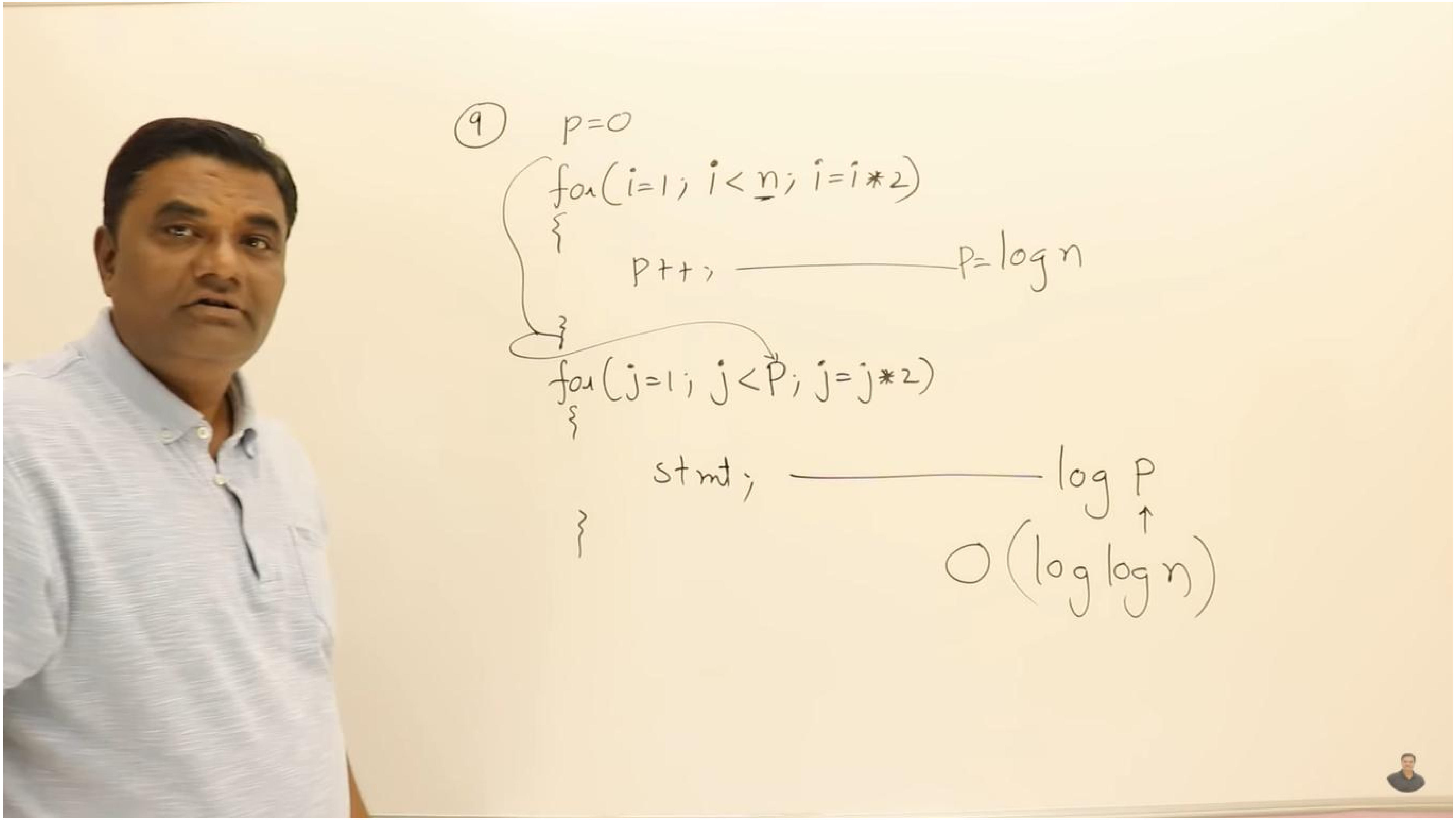
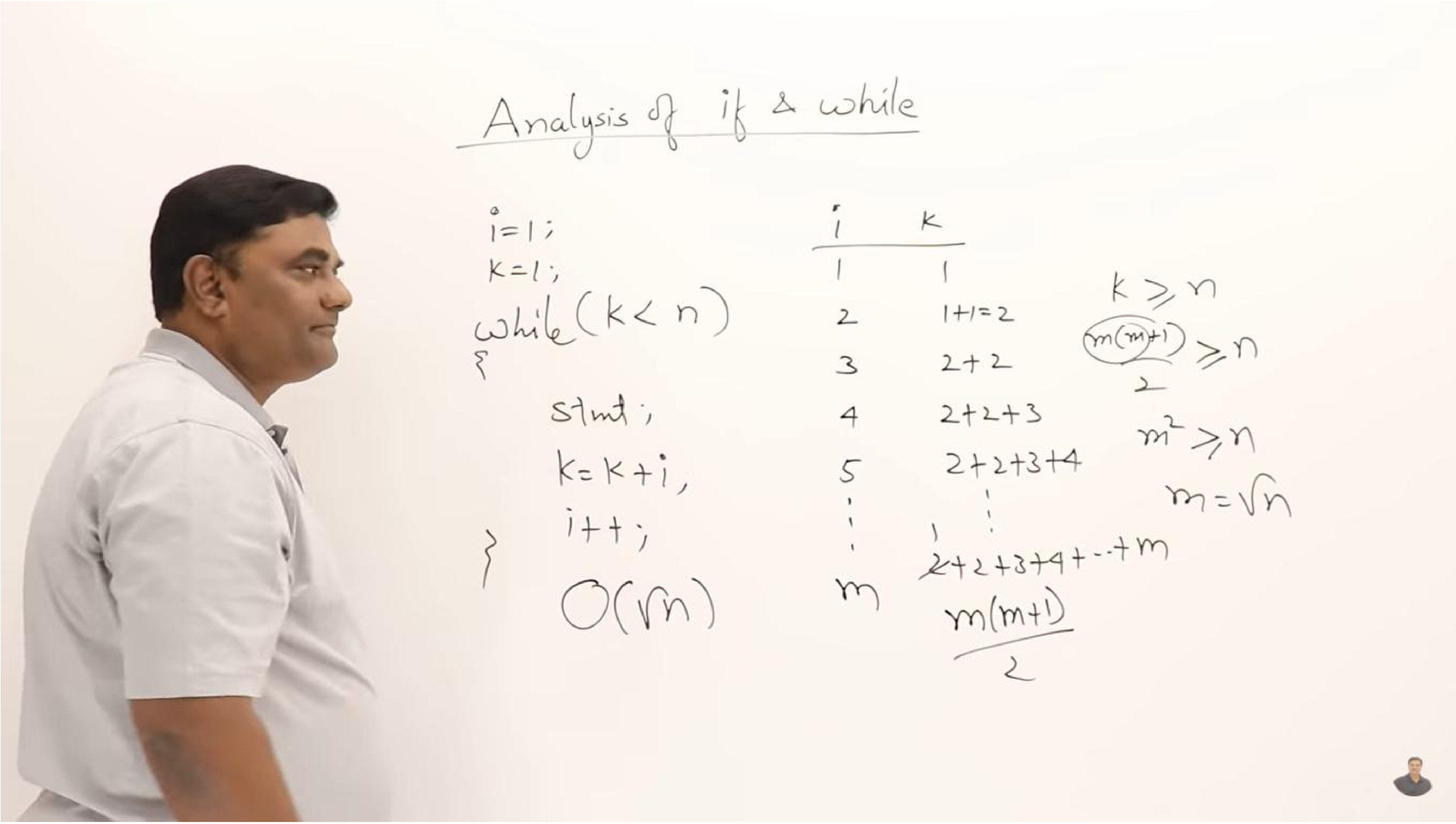
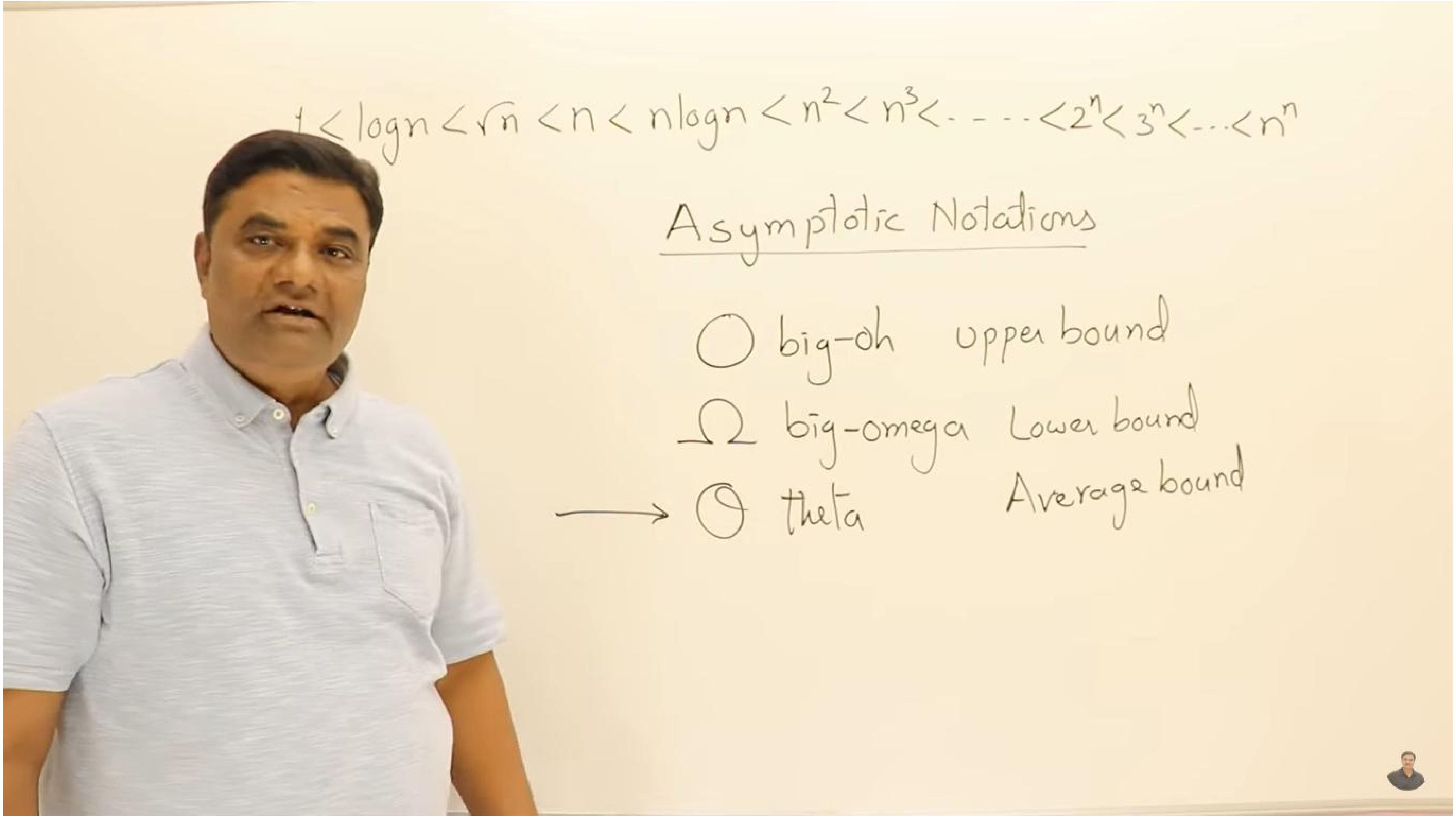
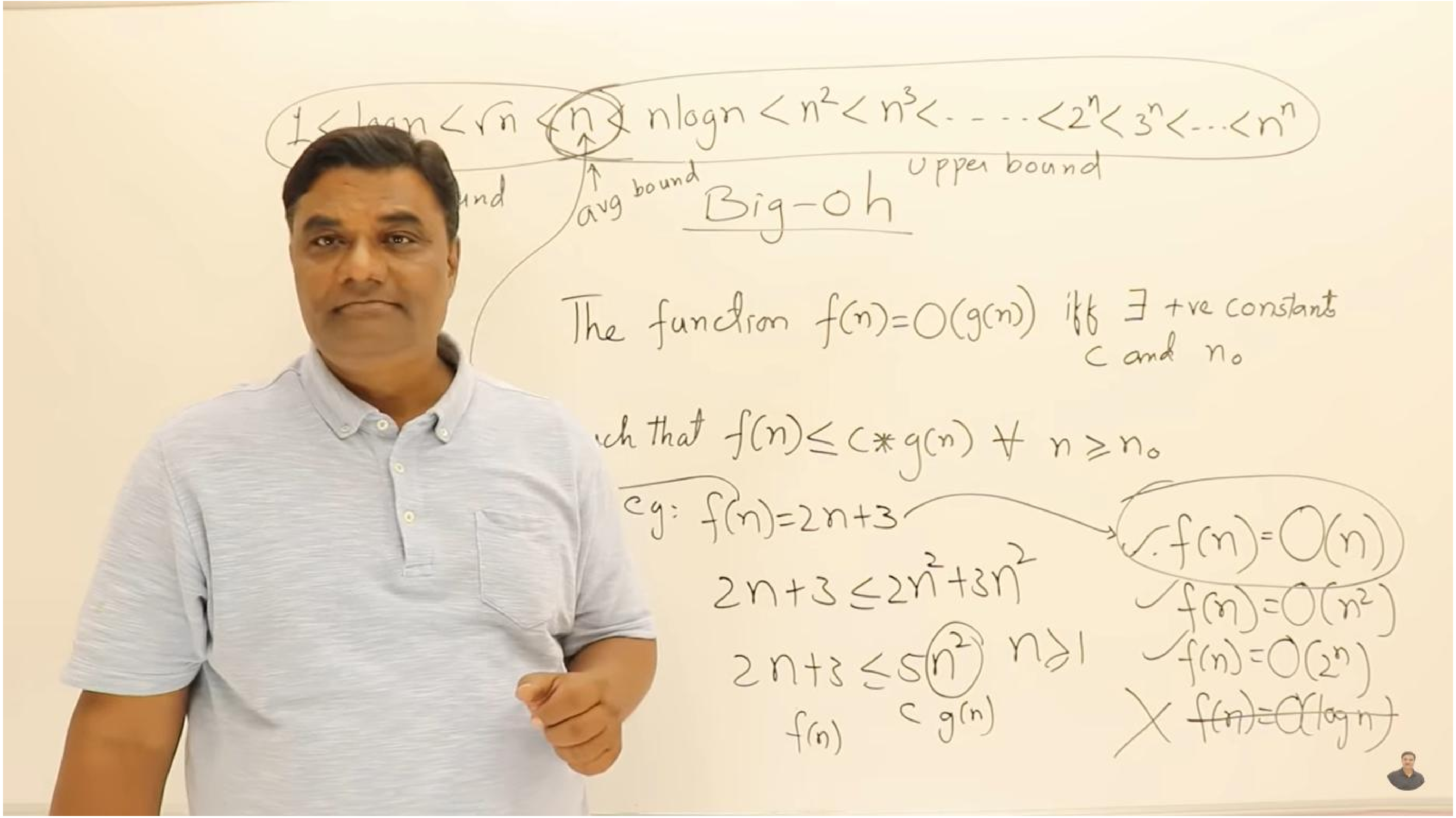


(4) p=0;
for(i=1; p<=n; i++)
{
p=p+i;
}
Assume p > n
∵ p = k(k+1)/2
k(k+1)/2 > n
k² > n
k > √n
i p
1 0+1=1
2 1+2=3
3 1+2+3
4 1+2+3+4
k 1+2+3+4+...+k
O(√n)


9
p=0
for(i=1; i<n; i=i*2)
{
p++;
p=log n
}
for(j=1; j<p; j=j*2)
{
stmt;
log p
}
O(log log n)


Analysis of if & while
i=1;
k=1;
while(k<n)
{
stmt;
k=k+i;
i++;
}
O(√n)
i k
1 1
2 1+1=2
3 2+2
4 2+2+3
5 2+2+3+4
m 2+2+3+4+...+m
m(m+1)/2
k ≥ n
m(m+1)/2 ≥ n
m² ≥ n
m=√n


$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 < \cdots < 2^n < 3^n < \cdots < n^n$
Asymptotic Notations
O big-oh upper bound
Ω big-omega Lower bound
Θ theta Average bound



1 < logn < √n < n < nlogn < n² < n³ < . . . . . < 2ⁿ < 3ⁿ < . . . < nⁿ
und
avg bound
upper bound
Big-Oh
The function f(n)=O(g(n)) iff ∃ +ve constants
c and n₀
ch that f(n) ≤ c * g(n) ∀ n ≥ n₀
eg: f(n)=2n+3
2n+3 ≤ 2n²+3n²
2n+3 ≤ 5n² n≥1
f(n) c g(n)
f(n)=O(n)
f(n)=O(n²)
f(n)=O(2ⁿ)
f(n)=O(log n)

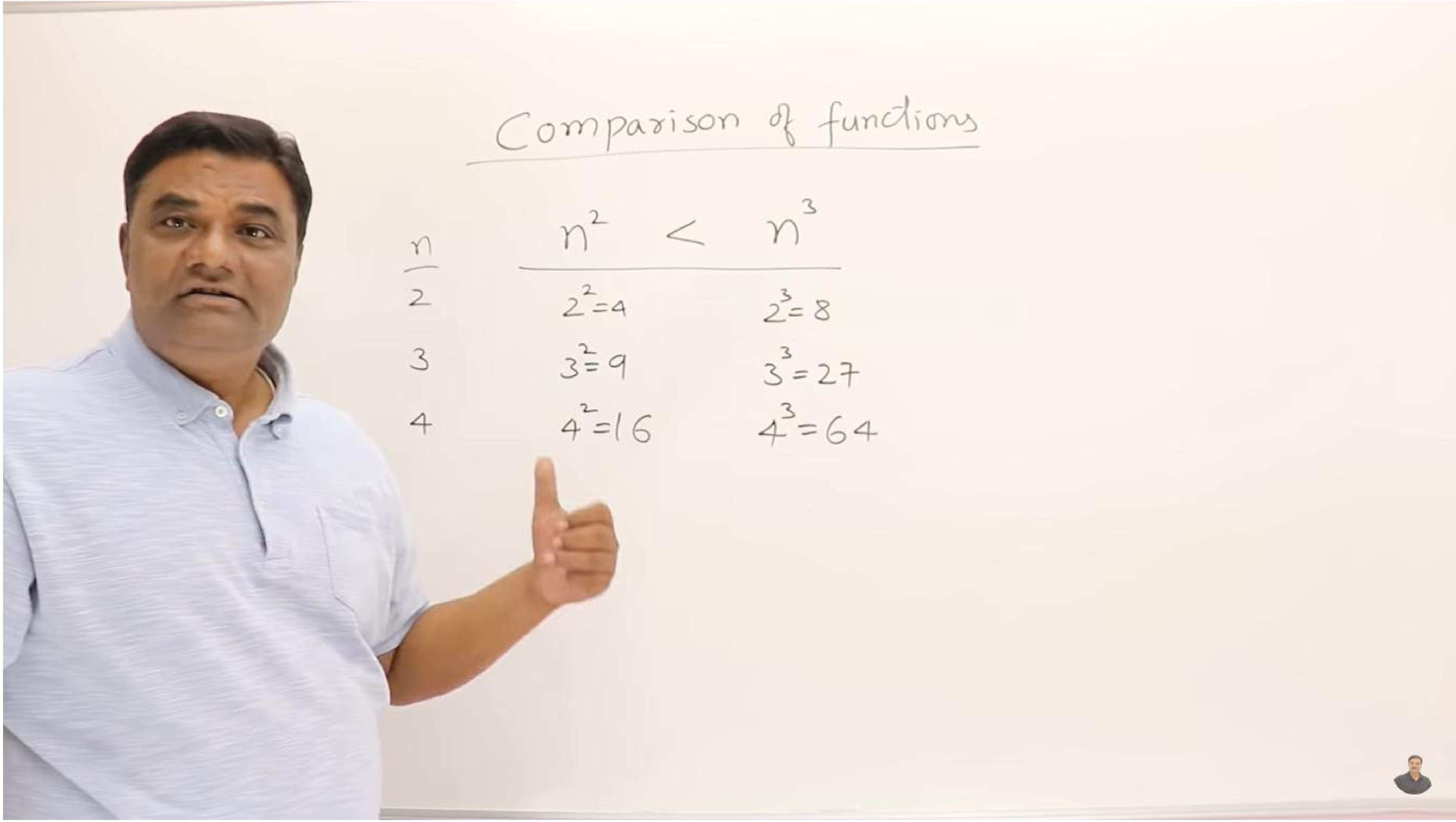
$$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 < \cdots < 2^n < 3^n \cdots < n^n$$

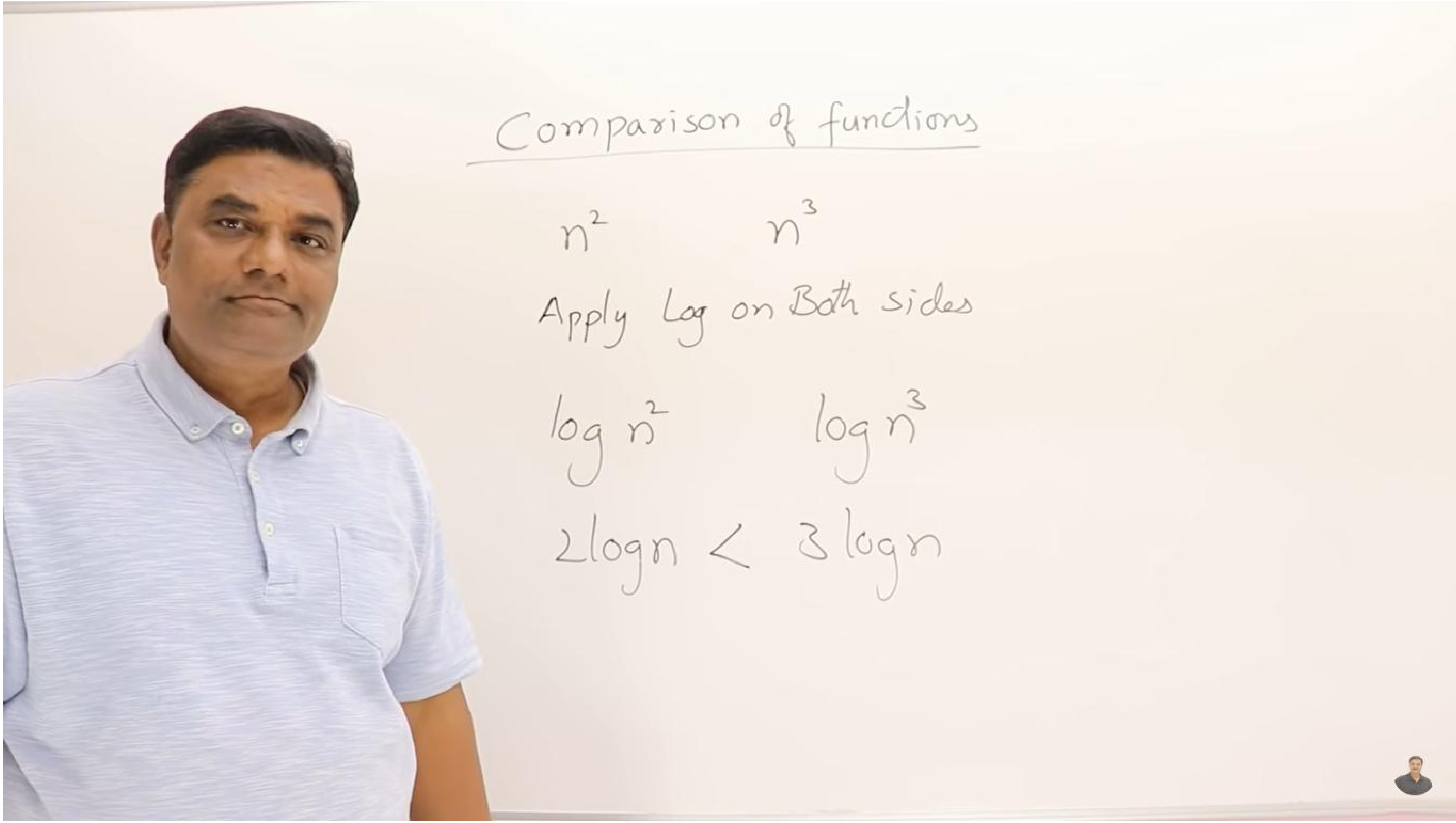
$$f(n) = n! = n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1$$

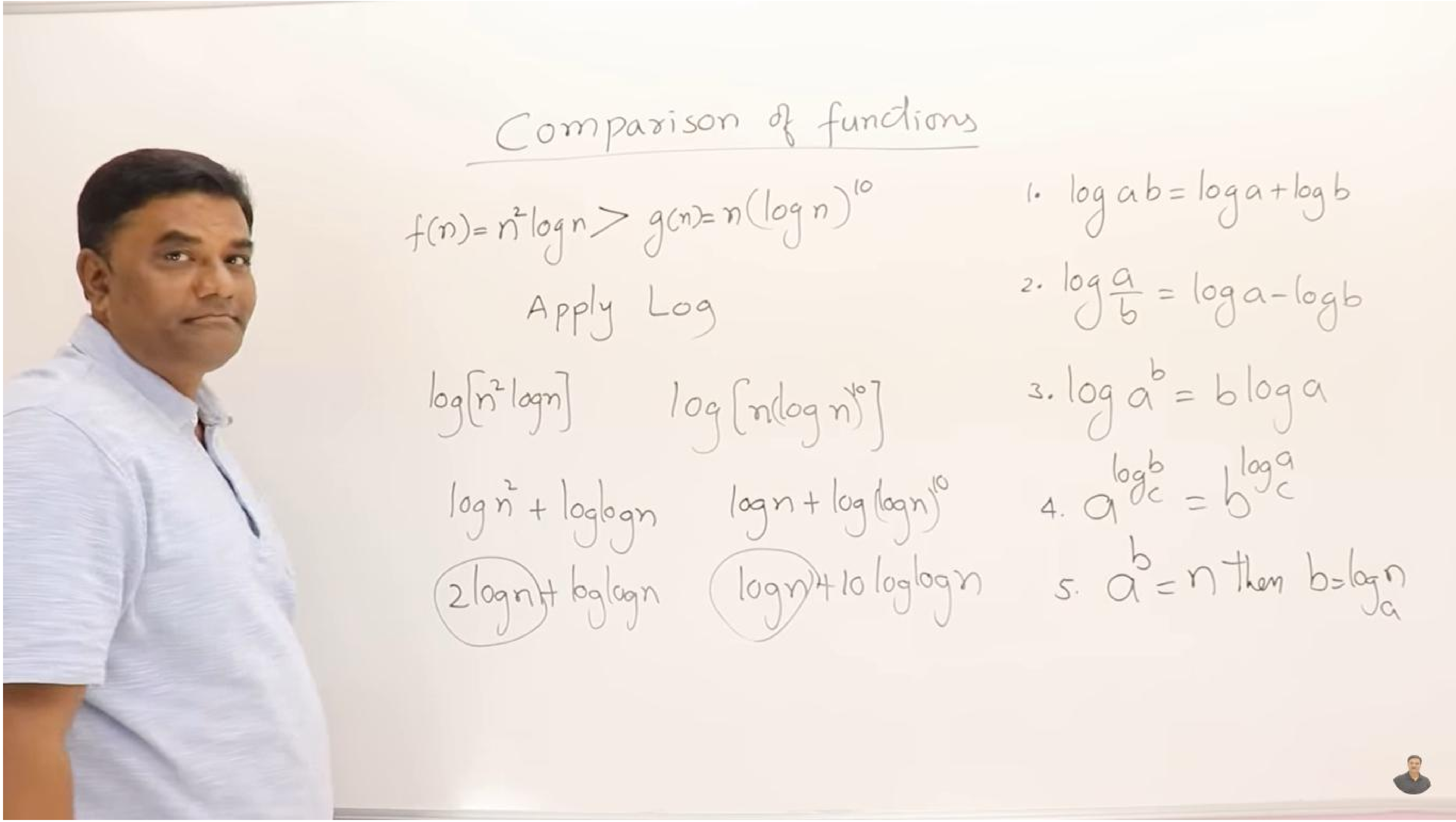
$$1 \times 1 \times 1 \times \cdots \times 1 \leq 1 \times 2 \times 3 \cdots \times n \leq n \times n \times n \times \cdots \times n$$

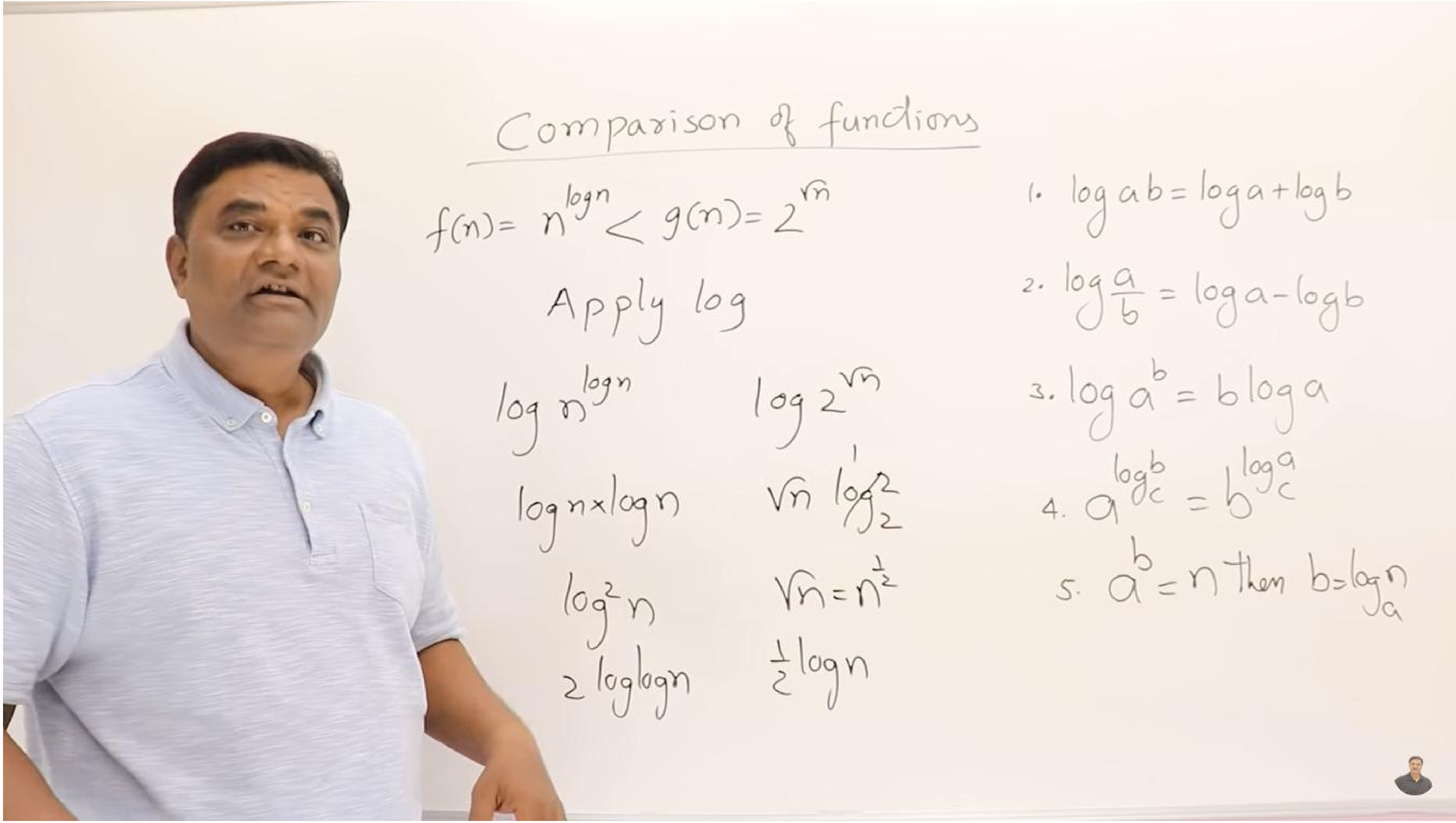
$$1 \leq n! \leq n^n$$

$$\Omega(1) \quad O(n^n)$$


Comparison of functions
n
$n^2 < n^3$
2
$2^2=4$
$2^3=8$
3
$3^2=9$
$3^3=27$
4
$4^2=16$
$4^3=64$


Comparison of functions
n^2 n^3
Apply Log on Both sides
$\log n^2$ $\log n^3$
$2\log n < 3\log n$


Comparison of functions
$f(n) = n^2 \log n > g(n) = n(\log n)^{10}$
Apply Log
$\log[n^2 \log n]$
$\log[n(\log n)^{10}]$
$\log n^2 + \log\log n$
$\log n + \log(\log n)^{10}$
$2\log n + \log\log n$
$\log n + 10 \log\log n$
1. $\log ab = \log a + \log b$
2. $\log \frac{a}{b} = \log a - \log b$
3. $\log a^b = b \log a$
4. $a^{\log_c b} = b^{\log_c a}$
5. $a^b = n$ then $b = \log_a n$


Comparison of functions
$f(n) = n^{\log n} < g(n) = 2^{\sqrt{n}}$
Apply log
$\log n^{\log n}$
$\log 2^{\sqrt{n}}$
$\log n \times \log n$
$\sqrt{n} \log_2 2$
$\log^2 n$
$\sqrt{n} = n^{\frac{1}{2}}$
$2 \log \log n$
$\frac{1}{2} \log n$
1. $\log ab = \log a + \log b$
2. $\log \frac{a}{b} = \log a - \log b$
3. $\log a^b = b \log a$
4. $a^{\log_c b} = b^{\log_c a}$
5. $a^b = n$ then $b = \log_a n$

Comparison of functions

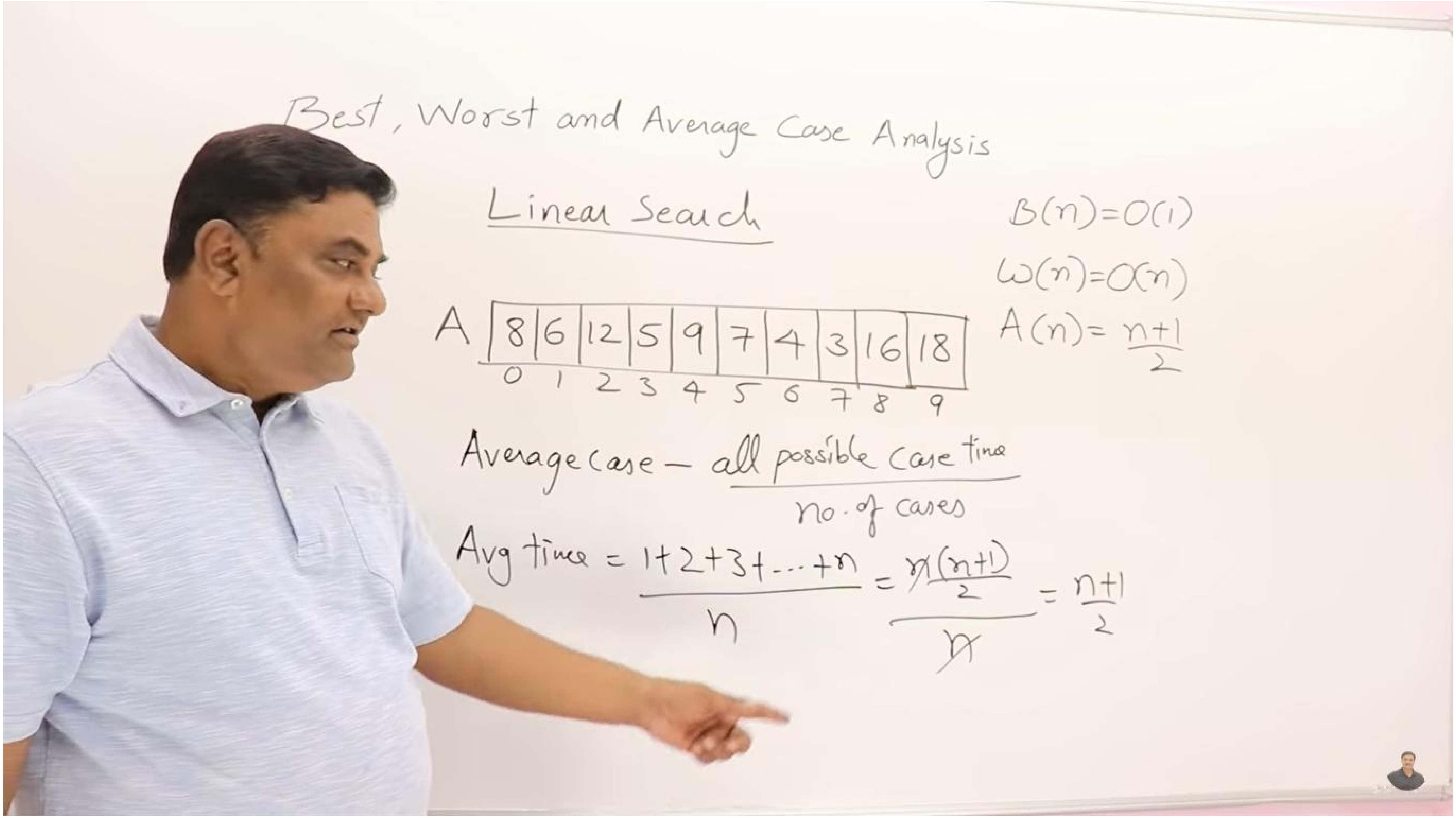
$$g1(n) = \begin{cases} n^3 & n < 100 \\ n^2 & n \geq 100 \end{cases}$$

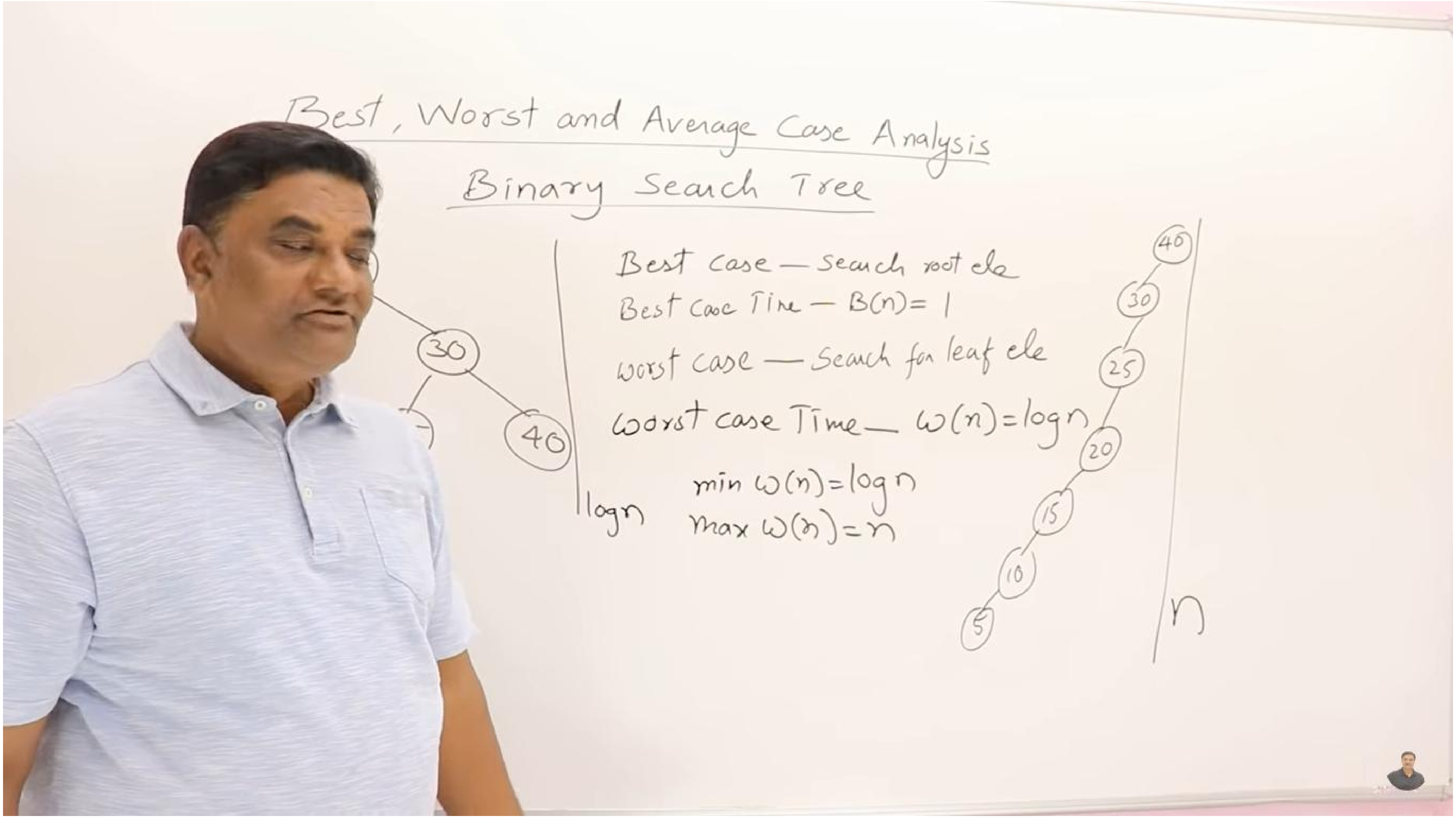
$$g2(n) = \begin{cases} n^2 & n < 10{,}000 \\ n^3 & n \geq 10{,}000 \end{cases}$$

$g2 > g1$

$n \rightarrow$ 100 $\quad n^2 \quad n^2 \quad$ 10,000 $\quad g2 > g1 \quad \infty$

$g1(n) > g2 \qquad g1 = g2$


Best, Worst and Average Case Analysis
Linear Search
A 8 6 12 5 9 7 4 3 16 18
0 1 2 3 4 5 6 7 8 9
B(n)=O(1)
W(n)=O(n)
A(n)= (n+1)/2
Average case – all possible case time / no. of cases
Avg time = (1+2+3+...+n)/n = (n(n+1)/2)/n = (n+1)/2


Best, Worst and Average Case Analysis
Binary Search Tree
Best case — search root ele
Best case Time — B(n) = 1
worst case — Search for leaf ele
worst case Time — w(n) = log n
min w(n) = log n
max w(n) = n
log n
n
30
40
25
20
15
10
5