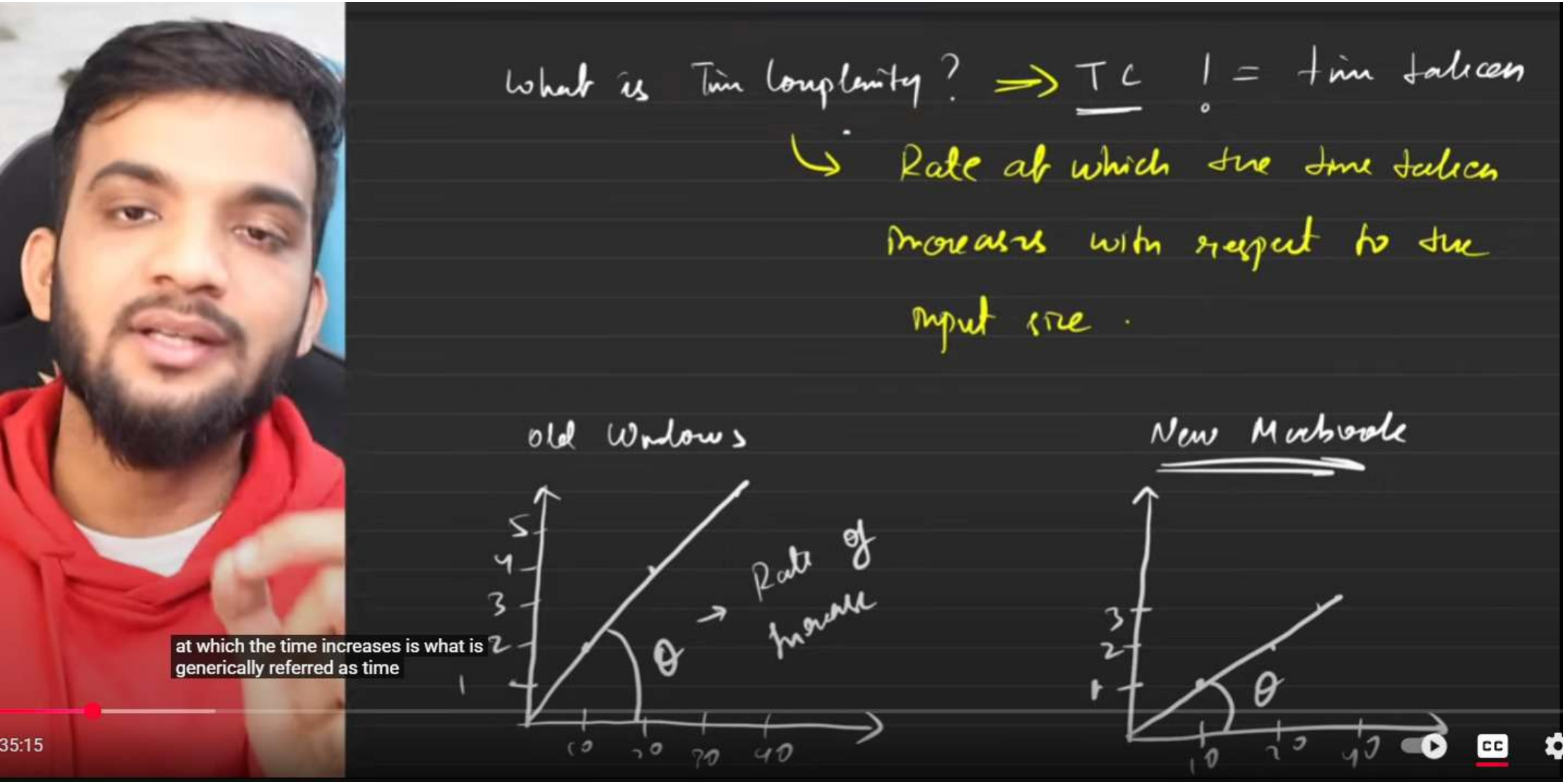
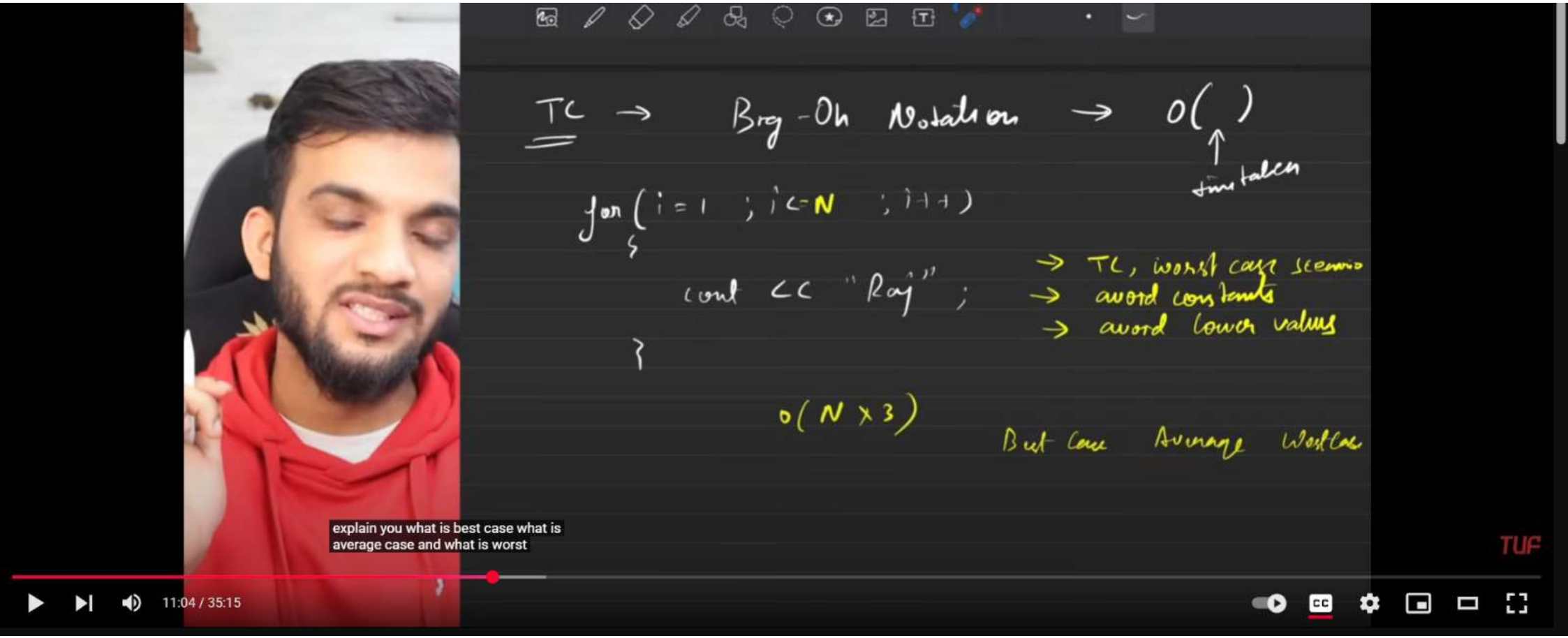
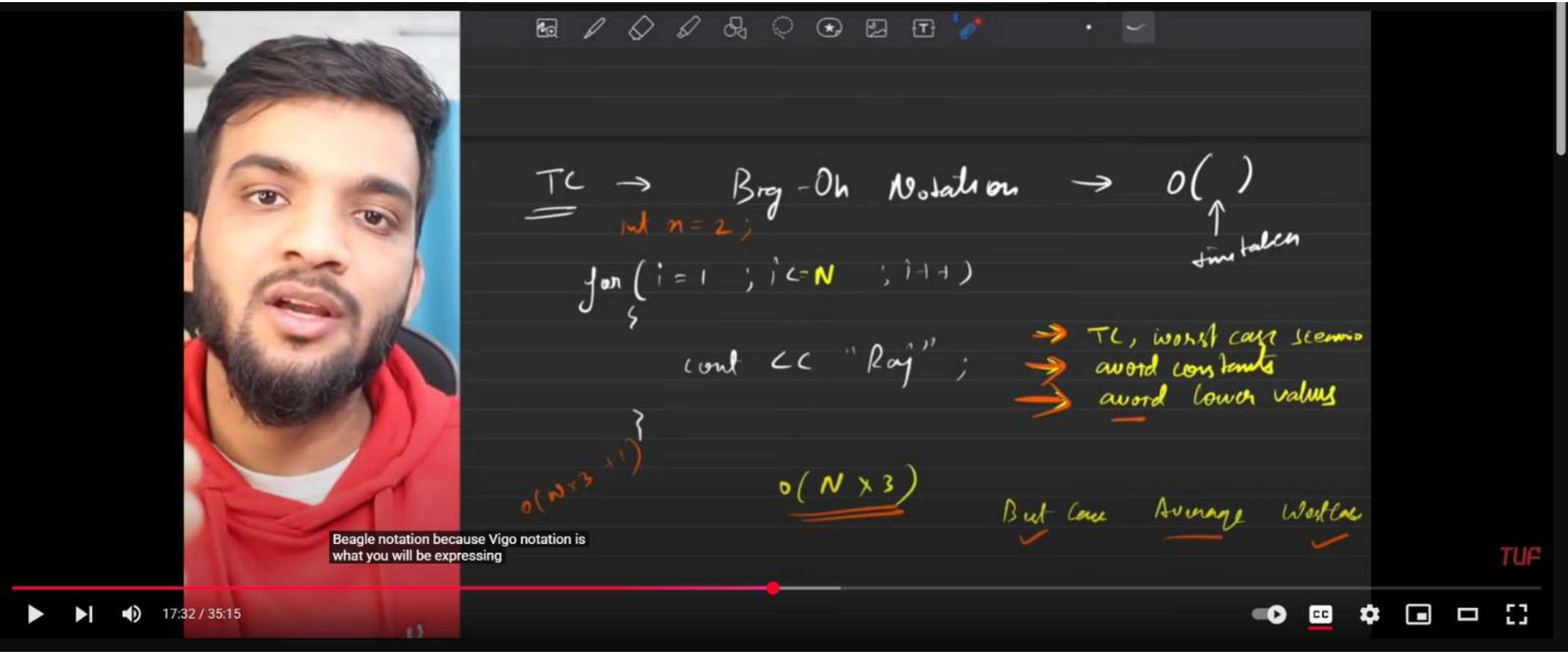
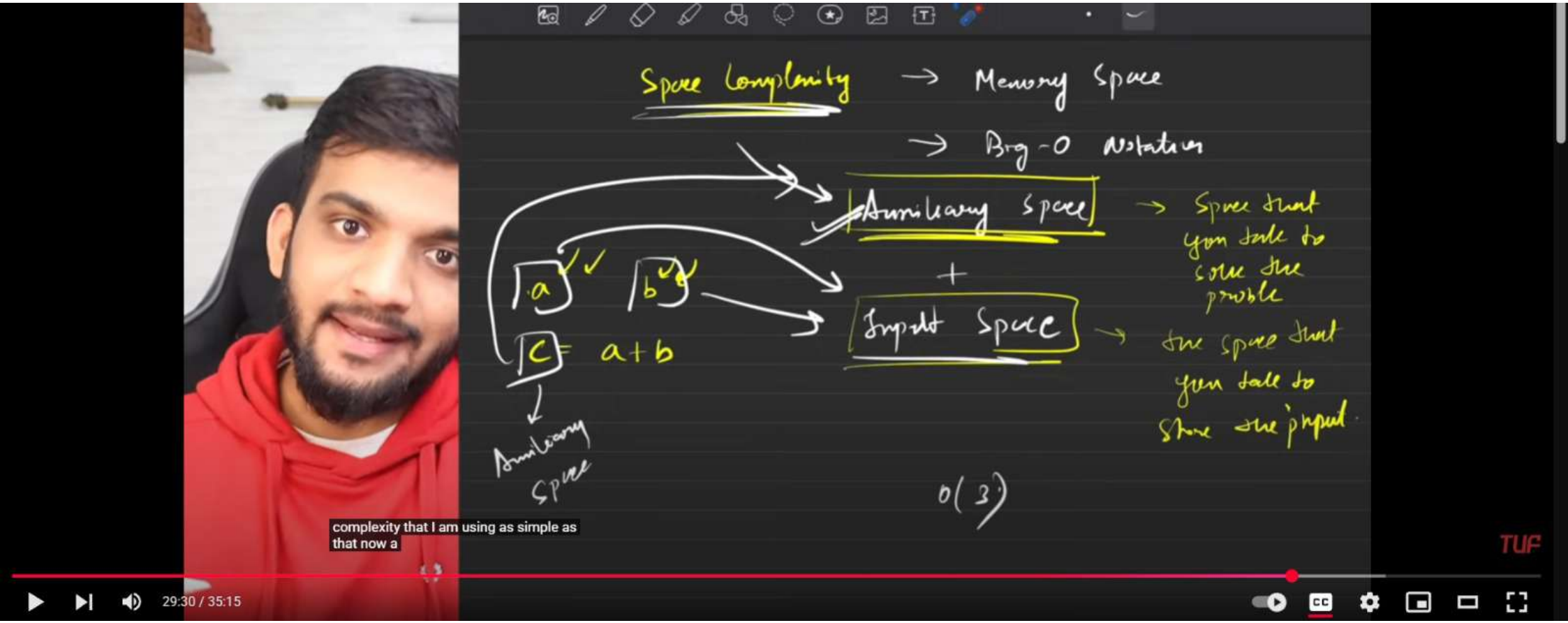
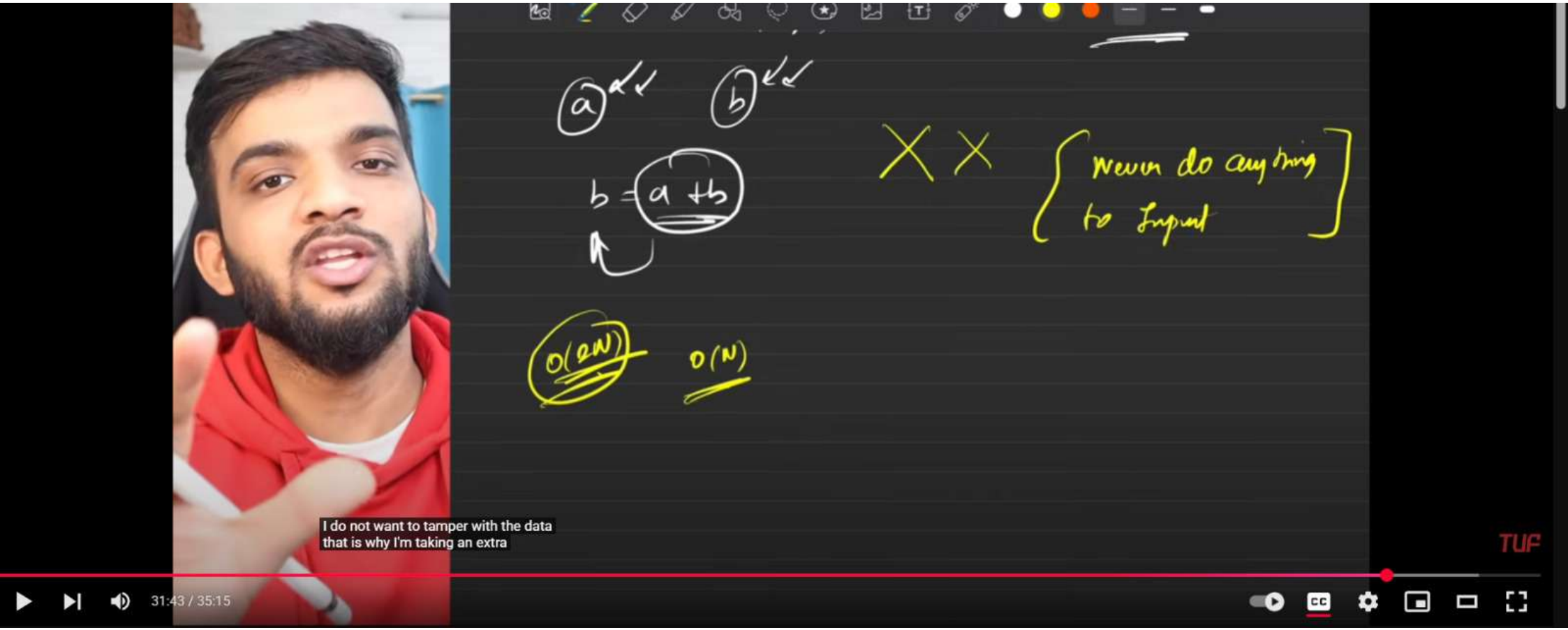
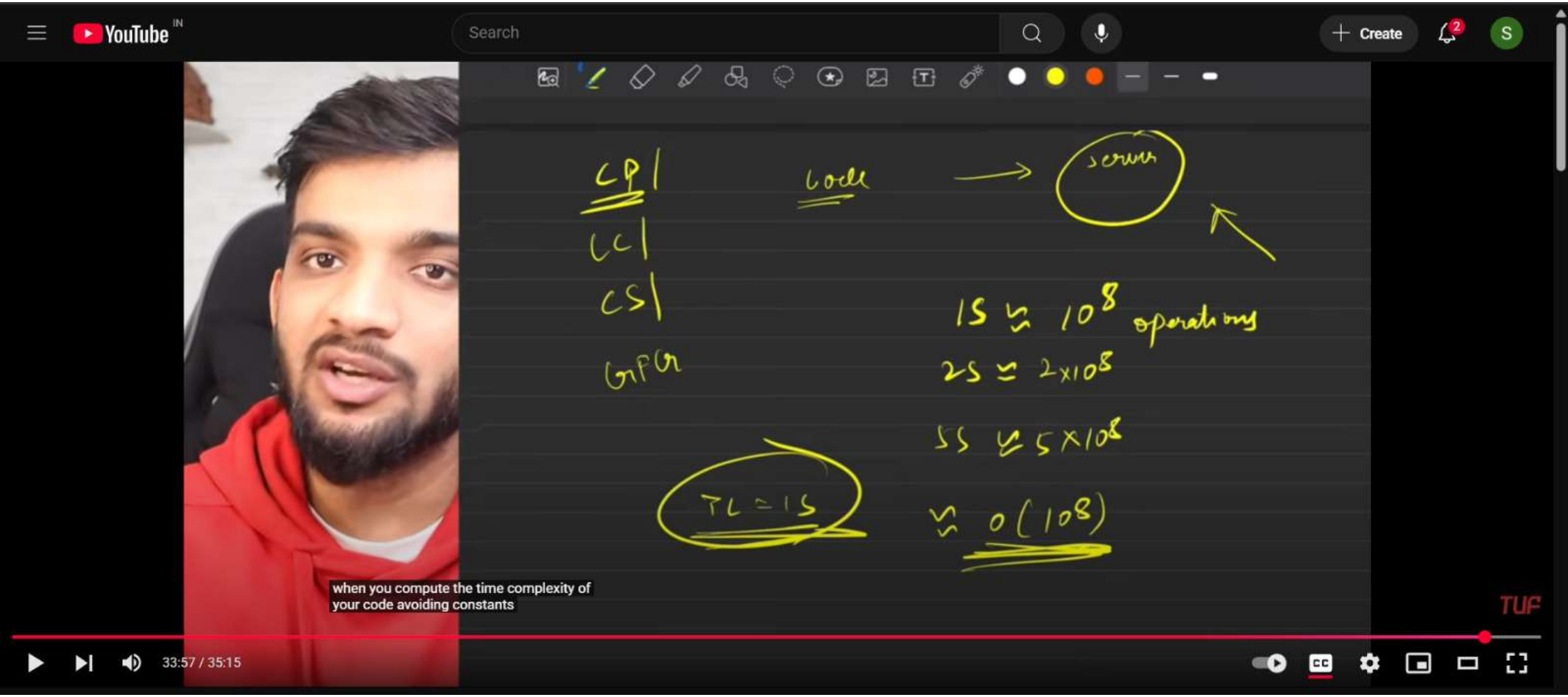

What is Time Complexity? => TC != time taken
Rate at which the time taken increases with respect to the input size.
Old Windows
New Machine
Rate of increase
θ
at which the time increases is what is generically referred as time
35:15
CC


TC → Big-Oh Notation → O( )
time taken
for (i = 1 ; i <= N ; i++)
{
cout << "Raj" ;
}
→ TC, worst case scenario
→ avoid constants
→ avoid lower values
O(N x 3)
Best Case Average WorstCase
explain you what is best case what is average case and what is worst
11:04 / 35:15
TUF


TC → Brg-Oh Notation → O( )
time taken
int n = 2;
for (i = 1 ; i <= N ; i++)
cout << "Raj" ;
TC, worst case scenario
avoid constants
avoid lower values
O(N x 3 + 1)
O(N x 3)
Best Case
Average
Worst Case
Beagle notation because Vigo notation is what you will be expressing
17:32 / 35:15
TUF


Space Complexity → Memory Space
→ Big-O Notation
Auxiliary Space → Space that you take to solve the problem
+
Input Space → the space that you take to store the input.
a
b
c = a+b
Auxiliary Space
O(3)
complexity that I am using as simple as that now a
TUF
29:30 / 35:15


a
b
b = a + b
O(2N)
O(N)
Never do anything to Input
I do not want to tamper with the data
that is why I'm taking an extra
31:43 / 35:15
TUF


YouTube IN
Search
Create
CPI
CCI
CSI
GFG
code
server
1S ≈ 10^8 operations
2S ≈ 2×10^8
5S ≈ 5×10^8
TL = 1S
≈ O(10^8)
when you compute the time complexity of your code avoiding constants
33:57 / 35:15
TUF