Give the order of growth (as a function of N) of the running times of each of the following code fragment:

```
int sum = 0;
for (int i = 1; i < N; i *= 2)
    for(int j = 0; j < i; j++)
        sum++;
```

1ST LOOP i varies $\Rightarrow$ 1, 2, 4, 8 .....

2ND LOOP j varies $\Rightarrow$ [0], [0,1], [0,1,2,3], [0,1,2.... 7], ...

order of growth = No of times (sum ++) is executed

$$= 1+2+4+8+ \ldots\ldots 2^{m-1}$$

where m is the number of elements in geometric progression.

$$\boxed{\begin{array}{l} N=9, \Rightarrow m=3 \\ N=33 \Rightarrow m=5 \end{array}}$$

order of growth $= \dfrac{a(1-r^m)}{1-r} = \dfrac{1\times 2^m - 1}{2-1} = \underline{2^m - 1}$

$-1\times$

but we know

$$N-1 = 2^{(m-1)} \Rightarrow 2(N-1) = 2^m$$

$\therefore$ order of growth $\Rightarrow 2(N-1) - 1$

$$= \underline{2N-3}$$

order of growth is linear in N