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 < N; j++)
      sum++;
```

1st loop i ⇒ 1, 2, 4, 8 .....

If N = 3 , i ⇒ 1, 2 ; count(i) = 2

N = 5 , i ⇒ 1, 2, 4 ; count(i) = 3

N = 33 , i ⇒ 1, 2, 4, 8, 16, 32 ; count(i) = 6

If $N = 2^x + 1$, i ⇒ 1, 2, 4, ... (N-1)

⇒ 1, 2, 4, ... 2^x

The count of i ⇒ x + 1

count(i) =

x+1 →

$2^x = (N-1)$

$x = \log(N-1)$

$x+1 = 1 + \log(N-1)$

for large(N) $(x+1) \cong \log N$

for long N

N-1 ≅ N

1+log N ≅ log N

2nd loop executes N times.

Total execution = $N * \log N \Rightarrow N \log N$