```
void function(int n)
{
   int count=0;
   for(int i=n/2; i<=n; i++)  ⟶ n times.
      for(int j=1; j<=n; 2*j)  ⟶ logn times.
         for(int k=1; k<=j; 2*k)  ⟶ loglogn times.
            count++;
}
```

The first for loop is running n times.

$i = 1, 2, 3, 4, 5 \ldots\ldots n/2 \rightarrow \frac{n}{2}$ times $\quad O(n/2) \in O(n)$

The second for loop connected first for loop and j value increases 2^a

$1, 2^1, 2^2, 2^3, 2^4 \ldots (2^k = n) \quad \log n = k \Rightarrow O(\log n)$

The third for loop connected first and second loop and k values increases 2^a

$1, 2^1, 2^2 \ldots 2^k = \log n \Rightarrow k = \log\log n \Rightarrow O(\log\log n)$

Time complexity $\Rightarrow n.\log n.(\log\log n)$