

# A Time Complexity Question - GeeksforGeeks

What is the time complexity of following function fun()? Assume that log(x) returns log value in base 2.

```
void fun()
{
    int i, j;
    for (i=1; i<=n; i++)
        for (j=1; j<=log(i); j++)
            printf("GeeksforGeeks");
}
```

Run on IDE

Time Complexity of the above function can be written as  $\Theta(\log 1) + \Theta(\log 2) + \Theta(\log 3) + \dots + \Theta(\log n)$  which is  $\Theta(\log n!)$

Order of growth of 'log n!' and 'n log n' is same for large values of n, i.e.,  $\Theta(\log n!) = \Theta(n \log n)$ . So time complexity of fun() is  $\Theta(n \log n)$ .

The expression  $\Theta(\log n!) = \Theta(n \log n)$  can be easily derived from following [Stirling's approximation \(or Stirling's formula\)](#).

$$\log n! = n \log n - n + O(\log(n))$$

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Sources:

[http://en.wikipedia.org/wiki/Stirling%27s\\_approximation](http://en.wikipedia.org/wiki/Stirling%27s_approximation)