Best case : - Ω

Worst case : - O

Average case : - theata …

1+ ½ +1/3 +1/4 + 1/5 +1/6 + 1/7 + 1/8 + 1/9 + 1/10 ……….. + 1/n < log(n)

i.e. always less than log(n)

as integra of 1/x ove 1 to n is log n it containg all the above + some other also …

for(int i=0;i<n;i++)

{

For(int j=0;j<n;j+=i)

{

//code

}

}

Complexity = n\*log(n)

<https://drive.google.com/file/d/1nGPYTKPCQS2rGb89E8lxaDREBi0ShUlc/view>

Comparison of functions on the basis of time complexity It follows the following order in case of time complexity: O(nn) > O(n!) > O(n3) > O(n2) > O(n.log(n)) > O(n.log(log(n))) > O(n) > O(sqrt(n)) > O(log(n)) > O(1) Note: Reverse is the order for better performance of a code with corresponding time complexity, i.e. a program with less time complexity is more efficient.