

Time Complexity of Loop with Powers

– By GeeksForGeeks

What is the time complexity of below function?

```
void fun(int n, int k)
{
    for (int i=1; i<=n; i++)
    {
        int p = pow(i, k);
        for (int j=1; j<=p; j++)
        {
            // Some O(1) work
        }
    }
}
```

Time complexity of above function can be written as $1^k + 2^k + 3^k + \dots n^k$.
Let us try few examples:

k=1

$$\begin{aligned}\text{Sum} &= 1 + 2 + 3 \dots n \\ &= n(n+1)/2 \\ &= n^2 + n/2\end{aligned}$$

k=2

$$\begin{aligned}\text{Sum} &= 1^2 + 2^2 + 3^2 + \dots n^2. \\ &= n(n+1)(2n+1)/6 \\ &= n^3/3 + n^2/2 + n/6\end{aligned}$$

k=3

$$\begin{aligned}\text{Sum} &= 1^3 + 2^3 + 3^3 + \dots n^3. \\ &= n^2(n+1)^2/4 \\ &= n^4/4 + n^3/2 + n^2/4\end{aligned}$$

In general, asymptotic value can be written as $(n^{k+1})/(k+1) + \Theta(n^k)$

Note that, in asymptotic notations like Θ we can always ignore lower order terms. So the time complexity is $\Theta(n^{k+1} / (k+1))$