

Time complexity of recursive function inside for loop

Asked 4 years, 6 months ago Modified 1 year, 4 months ago Viewed 6k times



If we have a function:-

2

```
int x=0;
```



```
int fun(int n)
{
```



```
    if(n==0)
        return 1;
```

1



```
    for(int i=0; i<n;i++)
        x += fun(i);
}
```

According to me, time complexity can be calculated as:-

$$T(n) = T(n-1) + T(n-2) + \dots + T(0).$$

$$T(n) = nT(n-1).$$

$$T(n) = O(n!).$$

Am I correct?

[time-complexity](#) [big-o](#)

Share Improve this question Follow

edited Aug 19, 2017 at 9:14



[Tony Tannous](#)

12.9k 8 44 79

asked Aug 19, 2017 at 5:47



[Zephyr](#)

1,401 3 17 35

2 Answers

Active

Oldest

Votes

If you're measuring the number of function calls (or additions -- it turns out the same), the correct recurrence relations are:

4

$$T(0) = 0$$

$$T(n) = T(0) + T(1) + T(2) + \dots + T(n-1) + n$$



You can compute the first few values:



$$T(0) = 0$$

$$T(1) = 1$$

$$T(2) = 3$$

$$T(3) = 7$$

$$T(4) = 15$$

You can guess from this that $T(n) = 2^n - 1$, and that's easy to check by a proof by induction.

In some sense you are right that $T(n) = O(n!)$ since $n! > 2^n$ for $n > 3$, but $T(n) = O(2^n)$ is a tighter bound.

Share Improve this answer Follow

answered Aug 19, 2017 at 9:18



Paul Hankin

48.8k

11

84

106

1 I think if we count the initial function call also then the recurrence would be :- $T(n) = T(n-1) + T(n-2) + \dots + 1$ and not n at the end. For $n=3$, $T(3) = T(0)+T(1)+T(2) + 1 = 1+2+4+1 = 8$. – [Zephyr](#) Aug 19, 2017 at 10:22

$$1. T(n) = T(n-1) + T(n-2) + T(n-3) + \dots + T(0)$$

4

// Replace n with $n-1$

$$2. T(n-1) = T(n-2) + T(n-3) + \dots + T(0)$$

Replace $T(n-2) + T(n-3) + \dots + T(0)$ with $T(n-1)$ in 1st Equation

```
T(n) = T(n-1) + T(n-1)
      = 2 * T(n-1)
      = 2 * 2 * T(n-2) // Using T(n-1) = 2 * T(n-2)
      = 2^n * T(n-n)
      = 2^n * T(0) // Consider T(0) = 1
      = 2^n
```

Share Improve this answer Follow

edited Sep 27, 2019 at 11:43

answered Sep 23, 2019 at 13:54



[Abhilekh Singh](#)

2,547 1 14 23