A Time Complexity Question

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");
}
```

Time Complexity of the above function can be written as
$\theta(\log 1) + \theta(\log 2) + \theta(\log 3) + ... + \theta(\log n)$ 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

Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion
- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

Writing code in comment? Please use **ideone.com** and share the link here.

17 Comments GeeksforGeeks

Sort by Newest

Join the discussion

Recursion
Geometric Algorithms

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing
Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

Devesh_D • 3 months ago

One more reason is there for use of quick sort more than merge sort because the cases (best , avg, worst) that means it is not depend on the input type(like quick sort if we are using randomized quick sort then we are sure that there is the time taken by the quick sort is always less than or equal to merge sort.

Prince • 8 months ago

Yes harsh, correct..n! 's upper bound is n^n so will beome nlogn.

Har?sh • a year ago

n!=1x2x3.....n
n!<=nxnx.....n
n!<=n^n
lgn!<=lgn^n
lgn!<=nlgn
T(n)=O(nlgn)(Asymptotic tight upperbound).

1

Guest ➚ Har?sh • 8 months ago

But here they are using Theta notation,but you have used Big-oh notati

Mihir • a year ago

I might be wrong, but we do not know anything about the complexity of the fun How can we comment about the overall complexity without knowing this? (Unl complexity. Then it is fine.)

2

Nidhi • a year ago

Shouldn't we consider the first loop "i" while calculating complexity ?

```
/* Paste your code here (You may delete these lines if not writing co
```

neha2210 ➦ Nidhi • a year ago

We are indeed considering the first loop also that is why we are adding
As you see if both loops are repeated for n times then the complexity w
it here in a similar manner.

vivek • 3 years ago

Does it mean that the given code is as effficient as following code?

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

Sandeep ➦ vivek • 3 years ago

@vivek: Both codes are of same time complexity as both are asympto

Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 29 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 32 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 57 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 59 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

take more time in general. As an example, both 2n and 10000n are of s
that takes 10000n time will definiely take more time. What we conclude
perfect, but it is the only availavle way to compare algorithms irrespect
architectures.. etc. As another example, we can consider Merge Sort a
complexity of Quick Sort is more and average case is same as Merge
of the hidden greater constants in Merge Sort.

Doom ➚ Sandeep • 3 years ago

@Sandeep: Could you please tell me more about the hidden gr
reason why we prefer quick sort over mergesort?

Bhagat Vishal ➚ Doom • 3 months ago

what i think , in case of deciding priority between the me
ammount of space taken by merge sort .but this could b
linked list without any extra space.

Amit ➚ Doom • 3 years ago

@Doom: These constants depends upon cost of execu
machine....

Like some machine incurs more cost in assignment tha

viresh ➚ Amit • 3 years ago

mergesort is not inPlace.. so each time u ask for a extra
there might be a page fault and also mergesort involves
makes mergesort costlier.. in contrast quicksort is inpla
changing the index rather than swapping the elements i
swaps 2s in d inner loop)..

Doom ➚ Amit • 3 years ago

@Amit: yes, you are right. But discussing about the arc
am more concerned about the specific instructions whi
expensive operation than quick sort.

Sandeep ➚ Doom • 3 years ago

@Doom: These constants are not fixed and vary from r
that the constants for MergeSort are greater than Quick

Sandeep ➚ Sandeep • 3 years ago

@Doom: These constants are greater if number of CPU
recursion are more. The CPU cycles might be used for
data movement etc.

For example, consider the following two loops. Time co
O(n), but the constants involved in Loop 2 are more tha
vary from machine to machine.

```
// Loop 1
for(i = 0; i < n; i++)
{
}

// Loop 2
for(i = 0; i < n; i++)
{
    printf("GeeksforGeeks");
```

```
}
```

Hope I made things clear this time.

Doom Sandeep • 3 years ago

@Sandeep: But what are these constants related to? I r
to point out? is it like the no. of comparisons involved? c
the array? plz give some examples about the constants