

Related Articles

Tail Call Elimination

Difficulty Level : Medium • Last Updated : 17 May, 2021

We have discussed (in [tail recursion](#)) that a recursive function is tail recursive if the recursive call is the last thing executed by the function.

C++

```
// An example of tail recursive function
void print(int n)
{
    if (n < 0)
        return;
    cout << " " << n;

    // The last executed statement is recursive call
    print(n-1);
}
```

Java

```
// An example of tail recursive function
static void print(int n)
{
    if (n < 0)
        return;

    System.out.print(" " + n);

    // The last executed statement
    // is recursive call
    print(n-1);
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Python3

```
# An example of tail recursive function
def print(n):

    if (n < 0):
        return

    print(" ", n)

    # The last executed statement is recursive call
    print(n - 1)

# This code is contributed by sanjoy_62
```

C#

```
// An example of tail recursive function
static void print(int n)
{
    if (n < 0)
        return;

    Console.Write(" " + n);

    // The last executed statement
    // is recursive call
    print(n - 1);
}

// This code is contributed by pratham76
```

Javascript

<script>

```
// An example of tail recursive function
function print(n)
{
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
// The last executed statement is recursive call
print(n-1);
}

</script>
```

We also discussed that a tail-recursive is better than a non-tail recursive as tail-recursion can be optimized by modern compilers. Modern compiler basically does **tail call elimination** to optimize the tail-recursive code.

If we take a closer look at the above function, we can remove the last call with goto. Below are examples of tail call elimination.

C++

```
// Above code after tail call elimination
void print(int n)
{
start:
    if (n < 0)
        return;
    cout << " " << n;

    // Update parameters of recursive call
    // and replace recursive call with goto
    n = n-1;
    goto start;
}
```

QuickSort : One more example

QuickSort is also tail recursive (Note that MergeSort is not tail recursive, this is also one of the reasons why QuickSort performs better)

C++

```
/* Tail recursive function for QuickSort
arr[] --> Array to be sorted,
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

```
if (low < high)
{
    /* pi is partitioning index, arr[p] is now
       at right place */
    int pi = partition(arr, low, high);

    // Separately sort elements before
    // partition and after partition
    quickSort(arr, low, pi - 1);
    quickSort(arr, pi + 1, high);
}
}
// See below link for complete running code
// http://geeksquiz.com/quick-sort/
```

The above function can be replaced by following after tail call elimination.

C++

```
/* QuickSort after tail call elimination
   arr[] --> Array to be sorted,
   low  --> Starting index,
   high --> Ending index */
void quickSort(int arr[], int low, int high)
{
    start:
    if (low < high)
    {
        /* pi is partitioning index, arr[p] is now
           at right place */
        int pi = partition(arr, low, high);

        // Separately sort elements before
        // partition and after partition
        quickSort(arr, low, pi - 1);

        // Update parameters of recursive call
        // and replace recursive call with goto
        low = pi+1;
        high = high;
        goto start;
    }
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Therefore job for compilers is to identify tail recursion, add a label at the beginning and update parameter(s) at the end followed by adding the last goto statement.

Function stack frame management in Tail Call Elimination :

Recursion uses a stack to keep track of function calls. With every function call, a new frame is pushed onto the stack which contains local variables and data of that call. Let's say one stack frame requires $O(1)$ i.e, constant memory space, then for **N** recursive call memory required would be $O(N)$.

Tail call elimination reduces the space complexity of recursion from $O(N)$ to $O(1)$. As function call is eliminated, no new stack frames are created and the function is executed in constant memory space.

It is possible for the function to execute in constant memory space because, in tail recursive function, there are no statements after call statement so preserving state and frame of parent function is not required. Child function is called and finishes immediately, it doesn't have to return control back to the parent function.

As no computation is performed on the returned value and no statements are left for execution, the current frame can be modified as per the requirements of the current function call. So there is no need to preserve stack frames of previous function calls and function executes in constant memory space. This makes tail recursion faster and memory-friendly.

Next Article:

[QuickSort Tail Call Optimization \(Reducing worst case space to \$\log n\$ \)](#)

This article is contributed by **Dheeraj Jain**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the [DSA Self Paced Course](#) at a student-friendly price and become industry ready. To complete your preparation from learning a language to DS Algo and many more, please refer [Complete Interview Preparation Course](#).

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

Like 34

Next

Tail Recursion

ADVERTISEMENT BY ADRECOVER

ADVERTISE ON GEEKSFORGEEKS

RECOMMENDED ARTICLES

Page : 1 2

01 QuickSort Tail Call Optimization (Reducing worst case space to $\log n$)
30, Mar 16

05 MSD(Most Significant Digit) Radix Sort
18, May 21

02 Tail Recursion
11, Mar 14

06 Implement Secure Hashing Algorithm - 512 (SHA-512) as Functional Programming Paradigm
03, May 21

03 Tail Call Elimination

07 Akra-Bazzi method for finding the

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

04 Tail recursion to calculate sum of array elements.

03, Sep 18

08 Brute Force Approach and its pros and cons

01, May 21

Article Contributed By :



Vote for difficulty

Current difficulty : [Medium](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [ishayadav181](#), [rutvik_56](#), [pratham76](#), [sanjoy_62](#), [famously](#)Article Tags : [Quick Sort](#), [tail-recursion](#), [Algorithms](#)Practice Tags : [Algorithms](#)

Improve Article

Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

[About Us](#)
[Careers](#)
[Privacy Policy](#)
[Contact Us](#)
[Copyright Policy](#)

Practice

[Courses](#)
[Company-wise](#)
[Topic-wise](#)
[How to begin?](#)

Learn

[Algorithms](#)
[Data Structures](#)
[Languages](#)
[CS Subjects](#)
[Video Tutorials](#)

Contribute

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !