Home Answer Spaces Notifications Search Quora Add Question or Link

Recursion Question That Contains Assumptions +3

What is tail recursion? Why is it so bad?

Answer Follow · 35 Request

Ad by Grammarly

Writing something important?

Join over 10 million Grammarly users and see what better, clearer writing can do for you. It's free!

Learn more at grammarly.com

6 Answers

Tikhon Jelvis, programming on my own and professionally for over a decade
Updated Dec 19, 2017

Tail recursion is a special kind of recursion where the recursive call is the *very last* thing in the function. It's a function that does not do anything *at all* after recursing.

This is important because it means that you can just pass the result of the recursive call through directly instead of waiting for it—you don't have to consume any stack space. A normal function, on the other hand, *has* to have a stack frame so that the compiler knows to come back to it (and have all the necessary variable values) after the recursive call is finished.

Some languages recognize this and implement "proper tail calls" or "tail call elimination": if they see a recursive call in tail position, they actually compile it into a jump that reuses the current stack frame instead of calling the function normally. This improves the memory usage of the function *asymptotically* and prevents it from overflowing the stack. With this behavior, tail recursion is actually generally a good thing: chances are you *do* want to write your functions in a tail-recursive form if you can.

Certain languages like Scheme promise proper tail calls in the standard, so you can always rely on them. In other languages like OCaml, the compiler gives you the same promise even if there is no formal language standard.

Tail recursion only starts being a problem in languages which *do not* promise to implement them properly. Some compilers, like GCC, offer it as an optimization. However, it isn't guaranteed, and you never want to rely on an optimization to make your code correct! (Remember: without proper tail calls, you could easily run out of stack space.) Other languages like Python don't even offer it as an optimization.

In these cases, you can always rewrite your tail recursive function as a loop—which are, presumably, guaranteed to run in constant space. Better yet, you can always switch to a language that *does* support proper tail calls ;).

86.1k Views · View Upvoters

Add a comment... Comment Recommended All

Upvote · 351 Share

There's more on Quora...

Pick new people and topics to follow and see the best answers on Quora.

Update Your Interests

Related Questions

- What is tail recursion? Are there any suitable examples to understand it?
- What is the difference between normal recursion and tail recursion with examples?
- Can all recursive functions be converted into tail recursive functions, and vice versa?
- Why is tail recursion less costly than normal recursion?
- What are some interesting applications of tail-recursion?
- Is recursion faster than loops?
- What is tail recursion and how to translate a tail recursion to a normal recursion?
- How does compiler know whether the recursion is a tail recursion or not and how does it optimize tail recursion?
- What is tail recursion?
- Is tail recursion a good thing or a bad thing?

+ Ask New Question

More Related Questions

In other languages

Dalam bahasa Indonesia: Apakah “tail recursive” buruk?

Question Stats

34 Public Followers

95,480 Views

Last Asked Feb 2, 2016

Edits