

Basics of recursion (Recursion)

Some programming languages ??allow a module or function to be called to itself. This technique is called Recursion.

What is Recursion?

Some programming languages ??allow a module or function to be called to itself. This technique is called Recursion. In recursion, a function can: call this function directly or call a function **b** that returns the call to the original function **a** . Function **a** is called recursive function.

For example , a function calls itself

```
int function ( int value ) { if ( value < 1 ) return ; function ( value - 1 )
```

For example , a function that calls another function returns a call to the original function

```
int function ( int value ) { if ( value < 1 ) return ; function ( value - 1 )
```

Characteristics of recursive functions

A recursive function can continue to happen countless times like an infinite loop. To avoid this, you must keep in mind the following two properties of the recursive function:

Basic condition : there must be at least one condition for when this condition is met, calling that function (called recursion) will stop.

Proximity: whenever the recursive function is called, it is asymptotic to the basic condition.

Recursive function deployment

Many programming languages ??implement recursion in the manner of stacks. In general, whenever a function (**caller** function) calls another function (**called function - callee**) or calls itself (callee), the caller function transmits the control to the callee. This transfer process may also include some data from caller to callee.

Compare recursion and loop

One might say that why using recursion while using a loop can do the same thing. The first reason is that recursion makes the program more readable and with today's improved CPU systems, recursion is much more

efficient when compared to loops.

Time complexity of recursive function

With loops, we take the number of loops to calculate the time complexity. Similar to recursion, assuming everything is constant, we calculate the time that a recursive call is created. A call made to a function will be $O(1)$, so for n is the time a recursive call is made, the time complexity of recursive function will be $O(n)$.

Memory complexity (complexity space) of recursive function

Memory complexity is estimated based on the amount of additional memory required for a module to be executed. With loops, the compiler almost doesn't need any extra memory. The compiler will update the value of the variable used right in the loop. But with recursion, the system needs to store dynamic logs every time a recursive call is made. Therefore it can be said that the memory complexity of the recursive function is higher than the loop.

According to Tutorialspoint

Previous article: Heap data structure

Next article: Problem of Hanoi Tower (Tower of Hanoi)

You finished reading the article "**Basics of recursion (Recursion)**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.