Iterative vs. Recursive Approaches

Eyal Lantzman, 5 Nov 2007 CPOL

4.08 (30 votes) Rate this:

Implication of not thinking of iterative solutions over recursive from performance (response time) point of view

Download source - 2.5 KB

Introduction

This article was originally posted at blogs.microsoft.co.il/blogs/Eyal.

Recursive function – is a function that is partially defined by itself and consists of some simple case with a known answer. Example: Fibonacci number sequence, factorial function, quick sort and more.
Some of the algorithms/functions can be represented in an iterative way and some may not.

Iterative functions – are loop based imperative repetitions of a process (in contrast to recursion which has a more declarative approach).

Comparison between Iterative and Recursive Approaches from Performance Considerations

Factorial

Hide Copy Code

```
//recursive function calculates n!
static int FactorialRecursive(int n)
{
    if (n <= 1) return 1;
    return n * FactorialRecursive(n - 1);
}

//iterative function calculates n!
static int FactorialIterative(int n)
{
    int sum = 1;
    if (n <= 1) return sum;
    while (n > 1)
    {
        sum *= n;
        n--;
    }
    return sum;
}
```

N	Recursive	Iterative
10	334 ticks	11 ticks
100	846 ticks	23 ticks
1000	3368 ticks	110 ticks
10000	9990 ticks	975 ticks

N	Recursive	Iterative
100000	stack overflow	9767 ticks

As we can clearly see, the recursive is a lot slower than the iterative (considerably) and limiting (stackoverflow).

The reason for the poor performance is heavy push-pop of the registers in the ill level of each recursive call.

Fibonacci

Hide Shrink Copy Code

```
//--------------- iterative version ---------------------
static int FibonacciIterative(int n)
{
    if (n == 0) return 0;
    if (n == 1) return 1;

    int prevPrev = 0;
    int prev = 1;
    int result = 0;

    for (int i = 2; i <= n; i++)
    {
        result = prev + prevPrev;
        prevPrev = prev;
        prev = result;
    }
    return result;
}

//--------------- naive recursive version ---------------------
static int FibonacciRecursive(int n)
{
    if (n == 0) return 0;
    if (n == 1) return 1;

    return FibonacciRecursive(n - 1) + FibonacciRecursive(n - 2);
}

//--------------- optimized recursive version ---------------------
static Dictionary<int> resultHistory = new Dictionary<int>();

static int FibonacciRecursiveOpt(int n)
{
    if (n == 0) return 0;
    if (n == 1) return 1;
    if (resultHistory.ContainsKey(n))
        return resultHistory[n];

    int result = FibonacciRecursiveOpt(n - 1) + FibonacciRecursiveOpt(n - 2);
    resultHistory[n] = result;

    return result;
}
```

N	Recursive	Recursive opt.	Iterative
5	5 ticks	22 ticks	9 ticks
10	36 ticks	49 ticks	10 ticks
20	2315 ticks	61 ticks	10 ticks
30	180254 ticks	65 ticks	10 ticks
100	too long/stack overflow	158 ticks	11 ticks
1000	too long/stack overflow	1470 ticks	27 ticks
10000	too long/stack overflow	13873 ticks	190 ticks
100000	too long/stack overflow	too long/stack overflow	3952 ticks

As before, the recursive approach is worse than iterative however, we could apply memorization pattern (saving previous results in dictionary for quick key based access), although this pattern isn't a match for the iterative approach (but definitely an improvement over the simple recursion).

Notes

1. The calculations may be wrong in big numbers, however the algorithms should be correct.

2. For timer calculations, I used `System.Diagnostics.Stopwatch`.

Points of Interest

1. Try not to use recursion in system critical locations.
2. Elegant solutions not always the best performing when used in "recursive situations".
3. If you required to use recursion, at least try to optimize it with dynamic programming approaches (such as memorization).

History

- Post: November 06, 2007

License

Share

About the Author

Eyal Lantzman
Software Developer (Senior) Taldor
Israel

In the last couple of years I'm working as Microsoft sub contractor in various project types - LOB, applications, CnC applications and Distributed applicaitons all of them considered to be extra large in tems of man power (or brain power), duration and geographic destribution between connected sites.

You may also be interested in...

A Decade of Cloud

SAPrefs - Netscape-like Preferences Dialog

Recursion made simple

Generate and add keyword variations using AdWords API

Code Iterations

Window Tabs (WndTabs) Add-In for DevStudio

Comments and Discussions

You must Sign In to use this message board.

Search Comments Go

First Prev Next

Nice Post
Member 11208991 19-Nov-14 6:58

It's a question of data structure
virtuPIC 15-Nov-07 20:45

blame C#, not .NET. F# does it better.
Jay R. Wren 13-Nov-07 2:00

Re: blame C#, not .NET. F# does it better.
ncarey 13-Nov-07 7:29

Hum
smesser 6-Nov-07 5:06

Re: Hum
Eyal Lantzman 6-Nov-07 20:08

Good Job
merlin981 6-Nov-07 4:24

Good Best Practice
The DevMan 6-Nov-07 0:33

Interesting
norm .net 5-Nov-07 21:46

Re: Interesting
Eyal Lantzman 5-Nov-07 22:32

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.