عنوان: **سری فیبوناچی و دات نت 4 !**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۰۹ ۲۱:۲۵:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: #C

سری معروف فیبوناچی که معرف حضور شما هست. سری از اعداد است که هر عدد آن مساوی حاصل جمع دو عدد ماقبل آن است. دو عدد اول این سری هم 0 و 1 هستند.
اگر بخواهیم این الگوریتم را به صورت یک متد بازگشتی نمایش دهیم به صورت زیر خواهد بود:

```
public static int Fibonacci(int x)
{
    if (x <= 1)
          return 1;
    return Fibonacci(x - 1) + Fibonacci(x - 2);
}
```

این الگوریتم چند مشکل دارد:
الف) برای اعداد بزرگ حتی با بکارگیری Int64 و یا double و امثال آن هم باز به جواب نخواهیم رسید (برای مثال 1500 را بررسی کنید).
ب) بسیار کند است.

در دات نت 4 برای کار با اعداد بزرگ، فضای نام System.Numerics معرفی شده است که حاوی نوع جدیدی از اعداد به نام BigInteger است.
اکنون اگر الگوریتم سری فیبوناچی را بر اساس این نوع داده جدید بازنویسی کنیم خواهیم داشت:

```
using System;
using System.Collections.Generic;
using System.Numerics; //needs a ref. to this assembly

namespace Fibonaci
{
   public class CFibonacci
   {
       public static int Fibonacci(int x)
       {
           if (x <= 1)
               return 1;
           return Fibonacci(x - 1) + Fibonacci(x - 2);
       }

       public static IEnumerable<BigInteger> BigFib(Int64 toNumber)
       {
           BigInteger previous = 0;
           BigInteger current = 1;

           for (Int64 y = 1; y <= toNumber; y++)
           {
               var auxiliar = current;
               current += previous;
               previous = auxiliar;
               yield return current;
           }
       }
   }
}
```

و مثالی در مورد نحوه استفاده از آن:

```
using System;
using System.Linq;
```

```
namespace Fibonaci
{
   class Program
   {
       static void Main()
       {
           foreach (var i in CFibonacci.BigFib(10))
           {
               Console.WriteLine("{0}", i);
           }

           var num = 12000;
           var fib = CFibonacci.BigFib(num).Last();
           Console.WriteLine("fib({0})={1}", num, fib);

           Console.WriteLine("Press a key...");
           Console.ReadKey();
       }
   }
}
```

برای نمونه با عدد 12000 خروجی برنامه در کسری از ثانیه (و نه چند دقیقه یا ساعت) به شرح زیر خواهد بود:

```
fib(12000)=514263424911336592579396579289954520826834443526829600435873863248622
654140207140138925514762610700100992755711440595791673560394372420894271363236890
022079562215696227914508914479059076682512326759880982463829024267831485466654040
473723840431646009452499112738578783466793628763574992042902850694420424444712000
522923293491036723024286623172850155258882103975837070714801788407729726923570540
718239988618967616871194346462509917026911008947695618108345420995773368214939050
416516589378605060670112152224358597976717485140234626345758771125412658570117230
314539904152316087295347817203811229658998710185320037352845593423725526271323000
638958253960120879480508550952336336384456686874402329262536204574599738895108380
235427851593712363899094709747385991667206113519035687818454094256246665597919120
022122897108388733347738351183912879567255044261504614219148441918105232576587700
998854927579270344092343400659284007697418021322038889294637023423241483436052750
289282804720944933596826625191272035818134041045429721812310762248914047306114590
033219426932250660389874831637094026012304670549443491110558503487799890585170690
960876267957092052157278434430545776800245076506782402407424212704226749074769270
224227339454507603236406191000216636750808704292990408918408807536464743300693320
723202183345822682199067634632613871613185005039704913147811005564943610633413710
435777879611831541255383712042967520284960846331034767830711777796040425810178880
282577849206596710823631711572896689043812540806768558155249875533726570636959700
396681091614491407072407112798594279199124438724052843058913668029547634219059700
152063114581874494201188387757074358579993108701995857608076801792582734610004600
975270649295645284743495470381783700438236289446709266019555376574271948158933650
884948631016675478967987281402249215848093553343797071563426205704968340863586920
309464672033306762062650479600723929916344563819984794114631821718163796501206840
350823997881370904601678190418455119512969342739887591698778395324922944303343280
469729051981315302242889228341251542112481598436096294690518890330853605407704800
256334512017053704475861775465777777593004101441661974393559036317730888125152150
096383779185952947478879700342090280194902103943924223024036870591194070058583790
521370989944572362900057457354208037588537232069921346429977050109405813861684270
473829736728167100146526325098889588516758942231174218294347289428786055699715120
652917833849101572036797794583542455798469738304725933701609775237079025751298030
072039857524154149354311250529579592001
```

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۸۹/۰۲/۰۹ ۲۲:۰۳:۲۳

استاد نمی دونم چقدر این سوال به موضوع ربط داره،ولی در بعضی جاها نوشته شده که حلقه foreach سنگین تر از حلقه for هستش و حلقه for سریعتر اجرا میشه.
بعضی ها هم میگن که اینطور نیست.می خواستم بدونم نظر شما در این مورد چی هست.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۰۹ ۲۲:۵۳:۱۵

بله. درسته. ولی من دراکثر موارد خوانایی کد را به چند میلی ثانیه بهبود کارآیی ترجیح می‌دم.

نویسنده: Meysam
تاریخ: ۱۳۸۹/۰۲/۱۰ ۱۰:۲۳:۱۴

مشکلی که روش اول داره فقط تو نوع داده ای نیست!

نویسنده: Nasser Hajloo
تاریخ: ۱۳۸۹/۰۲/۱۱ ۱۱:۵۱:۴۸

بخوبی یادم میاد که در درس طراحی الگوریتم خونده بودیم که نوشتن دنباله فیبوناچی بصورت بازگشتی صحیح نیست و بهتره که از روش های ساده یعنی استفاده از یک حلقه ساده برای بدست آوردن جواب استفاده کرد. که شما هم درواقع برنامه رو بصورت یک حلقه ساده نوشتید و با yield مقدار نهایی رو برگردوندید. کار بسیار خوبیه و از شما سپاسگذارم که این موارد رو یادآوری می کنید. اینها نشون میده که نوشتن یک متد بصورت اشتباه چقدر کارایی برنامه رو پایین میاره. امیدوارم در آینده الگوریتم های دیگری رو هم بررسی کنید.

نویسنده: ramin
تاریخ: ۱۳۸۹/۰۲/۱۲ ۱۳:۰۸:۵۹

سلام
من VS2010 رو نصب کردم ولی فضای نام System.Numerics رو نمیشناسه.
مشکل از چی میتونه باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۱۲ ۱۴:۲۷:۲۵

نوشتم که ... needs a ref. to this assembly
از منوی پروژه گزینه‌ی add reference ، ارجاعی را به اسمبلی استاندارد System.Numerics اضافه کنید.

نویسنده: MHM5000
تاریخ: ۱۳۸۹/۰۲/۱۳ ۱۶:۳۹:۰۸

سلام استاد...
همونطوری که به ویکیپدیا لینک داده بودید، برای روش دنباله فیبوناچی روش زیر هم وجود دارد:
http://albums.kimag.es/albums/mhm5000/71051717.jpg
اگه از این فرمول استفاده بشه، جواب سریعتر از حالتی که الآن شما توضیح دادید، بدست میاد؟
برای اعداد بزرگتر عرض می کنم...

جدیدا VB کار شدم اونم از ورژن ۶!
می خواستم بدونم بهتره شروع به یادگیری 2010 بکنم(یک ضرب) یا همون 6 خوبه؟ اصلا فرق محسوسی داره؟
چون من قابلیتی مثل برنامه حاظر توی vb6 ندیدم که بعد از عدد شانزدهم رو نمایش بده!
ممنون
MHM5000 | Erudite.ir

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۲/۱۳ ۱۹:۲۴:۰۵

سلام
- تمرکز این مطلب روی بهینه سازی الگوریتم نبود. بیشتر قصد داشتم اعداد عظیم الجثه رو معرفی کنم که برای مثال در این یک نمونه کاربرد دارد یا مباحث رمزنگاری اطلاعات و الگوریتم RSA
- با VB.Net شروع کنید. ساپورت VB6 احتمالا در محصولات سری بعد مایکروسافت به پایان می‌رسد یا بسیار محدود خواهد بود:
http://msdn.microsoft.com/en-us/vbrun/ms788708.aspx

نویسنده: SpEEdY
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۰:۱۹:۰۲

من یه سوال دارم. امکان اینکه از bigInteger تو c++ استفاده بشه نیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۰:۲۹:۰۷

از کتابخانه OpenSSL در CPP استفاده کنید:
http://www.openssl.org/docs/crypto/bn.html
dll یا حتی lib آماده آن برای ویندوز هم هست:
http://www.slproweb.com/products/Win32OpenSSL.html

نویسنده: SpEEdY
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۰:۵۰:۳۰

از کتابخانه OpenSSL در CPP استفاده کنید:

ممنون بابت جوابتون. راستش من خیلی مبتدی هستم. یه الگوریتم دارم که بر طبق اون باید اعداد تصادفی خیلی بزرگ تولید کنم، اونها رو جمع و ضرب کنم. اینکه چطوری باید از dll یا lib استفاده کنم رو بلد نیستم. از VS2008 استفاده میکنم. اگر لطف کنین و بیشتر توضیح بدین که چیکار کنم ممنون میشم.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۵:۳۴:۵۴

پاسخ:
http://www.dotnettips.info/2010/08/openssl.html