MAXimal

home
algo
bookz
forum
about

added: 10 Jun 2008 17:59
edited: 28 Aug 2011 23:42

Fibonacci numbers

Contents [hide]

Definition

The Fibonacci sequence is defined as follows:

$$F_0 = 0,$$
$$F_1 = 1,$$
$$F_n = F_{n-1} + F_{n-2}.$$

Several of its first members:

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ ,\ 55,\ 89,\ \ldots$$

History

These figures were introduced in 1202 by Leonardo Fibonacci (also known as Leonardo Pisano). However, it was thanks to the mathematician of the 19th century, Lucas, that the name "Fibonacci numbers" became common.

However, Indian mathematicians mentioned the numbers of this sequence even earlier: Gopala until 1135, Hemachandra in 1150.

Fibonacci numbers in nature

Fibonacci himself mentioned these figures in connection with this task: "A man planted a couple of rabbits in a pen surrounded on all sides by a wall.How many pairs of rabbits a year this pair can produce if it is known that every month, starting from the second, each pair rabbits produce one pair? ". The solution of this problem will be the numbers of the sequence, now called in his honor. However, the situation described by Fibonacci is more a game of reason than real nature.

Indian mathematicians Gopal and Hemachandra mentioned the numbers of this sequence in connection with the number of rhythmic patterns that result from the alternation of long and short syllables in verse or strong and weak parts in music. The number of such figures, having a total $n$fraction, is F_n.

Fibonacci numbers also appear in the work of Kepler in 1611, who thought about the numbers occurring in nature (the work "On hexagonal snowflakes").

An interesting example of a plant is the yarrow, in which the number of stems (and hence the flowers) is always a Fibonacci number. The reason for this is simple: after being initially with a single stem, this stem is then divided into two, then another one branches off from the main stem, then the first two stems again branch, then all the stems except the two last branch out, and so on. Thus, each stem after its appearance "skips" one branch, and then starts to divide at each level of branching, which gives the result of Fibonacci numbers.

Generally speaking, in many colors (for example, lilies), the number of petals is one or another Fibonacci number.

Also in botany, the phenomenon of phyllotaxis is known. An example is the location of the sunflower seeds: if you look at them from above, you can see at the same time two series of spirals (as if superimposed on each other): one is twisted clockwise, the other is against. It turns out that the number of these spirals roughly coincides with two consecutive Fibonacci numbers: 34 and 55 or 89 and 144. Similar facts are true for some other colors, as well as for pine cones, broccoli, pineapples, etc.

For many plants (according to some data, for 90% of them), such an interesting fact is also true. Consider a leaf, and we will go down from it until we reach the leaf located on the stem in exactly the same way (ie, directed exactly to the same side). In passing, we will consider all the leaves that came across to us (ie located in height between the starting sheet and the final one), but located differently. Numbering them, we will gradually make turns around the stem (since the leaves are located on the stalk in a spiral). Depending on whether you turn clockwise or counterclockwise, a different number of turns will result. But it turns out that the number of turns that we have made in a clockwise direction, the number of turns counterclockwise, and the number of the leaves found form 3 consecutive Fibonacci numbers.

However, it should be noted that there are plants for which the above calculations will give numbers from very different sequences, so you can not say that the phenomenon of phyllotaxis is a law - it is more of an entertaining trend.

Properties

Fibonacci numbers have a lot of interesting mathematical properties.

Here are just a few of them:

- Cassini's ratio:

  $$F_{n+1} F_{n-1} - F_n^2 = (-1)^n.$$

- Rule "addition":

  $$F_{n+k} = F_k F_{n+1} + F_{k-1} F_n.$$

- From the preceding equality it $k = n$follows that:

  $$F_{2n} = F_n (F_{n+1} + F_{n-1}).$$

- From the preceding equality we can obtain by induction that

  F_{nk}always a multiple F_n.
- The converse to the preceding statement is also true:

  if $F_m$multiple F_n, then $m$multiple n.
- GCD equality:

  $$\gcd(F_m, F_n) = F_{\gcd(m,n)}.$$

- With respect to Euclid's algorithm, Fibonacci numbers have the remarkable property that they are the worst input data for this algorithm (see "Lame Theorem" in Euclid's Algorithm).

Fibonacci number system

Tsekendorff's theorem asserts that any natural number $n$can be represented uniquely in the form of a sum of Fibonacci numbers:

$$N = F_{k_1} + F_{k_2} + \ldots + F_{k_r}$$

where $k_1 \geq k_2 + 2, k_2 \geq k_3 + 2, \cdots, k_r \geq 2$(i.e., in the recording can not use two adjacent Fibonacci numbers).

It follows that any number can be uniquely written in the **Fibonacci number system** , for example:

$$9 = 8 + 1 = F_6 + F_1 = (10001)_F,$$
$$6 = 5 + 1 = F_5 + F_1 = (1001)_F,$$
$$19 = 13 + 5 + 1 = F_7 + F_5 + F_1 = (101001)_F,$$

and in no number can not go two units in a row.

It is not difficult to obtain the rule of adding one to the number in the Fibonacci number system: if the lowest digit is 0, then it is replaced by 1, and if equal to 1 (that is, at the end is 01), then 01 is replaced by 10. Then "correct" recording, consistently correcting everywhere 011 by 100. As a result, in a linear time, a new number will be recorded.

The conversion of the number to the Fibonacci number system is carried out by a simple "greedy" algorithm: simply sort through the Fibonacci numbers from large to smaller and, if some $F_k \leq n$, $F_k$enter the number record n, and we take it $F_k$away $n$and continue the search.

The formula for the n-th Fibonacci number

The formula through radicals

There is a remarkable formula called by the name of the French mathematician Binet, although she was known before him by Moivre:

$$F_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}.$$

This formula is easy to prove by induction, but it can be derived by means of the notion of generating functions or by solving a functional equation.

You can immediately notice that the second term is always less than 1 modulo, and moreover, it decreases very rapidly (exponentially). Hence it follows that the value of the first term gives "almost" value F_n. This can be written in strict form:

$$F_n = \left[\frac{\left(\frac{1+\sqrt{5}}{2}\right)^n}{\sqrt{5}}\right],$$

where the square brackets denote rounding to the nearest integer.

However, for practical use in calculations, these formulas are of little use, because they require very high accuracy in working with fractional numbers.

Matrix formula for Fibonacci numbers

It is easy to prove the following matrix equality:

$$\begin{pmatrix} F_{n-2} & F_{n-1} \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} F_{n-1} & F_n \end{pmatrix}.$$

But then, denoting

$$P \equiv \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

we obtain:

$$\begin{pmatrix} F_0 & F_1 \end{pmatrix} \cdot P^n = \begin{pmatrix} F_n & F_{n+1} \end{pmatrix}.$$

Thus, in order to find the n5th Fibonacci number, we must raise the matrix $P$to a power n.

Recalling that the matrix $n$can be raised to the ith power $O(\log n)$(see Binary exponentiation), it turns out that the n-th Fibonacci number can be easily calculated $O(\log n)$using only integer arithmetic.

Periodicity of the Fibonacci sequence modulo

Consider the Fibonacci sequence $F_i$by some modulus p. Let us prove that it is periodic, and the period begins with $F_1 = 1$(that is, the pre- period contains only F_0).

Let us prove this by contradiction. Consider the $p^2 + 1$pair of Fibonacci numbers taken modulo p:

$$(F_1, F_2),\ (F_2, F_3),\ \ldots,\ (F_{p^2+1}, F_{p^2+2}).$$

Since $p$only p^2different pairs can be modulo , there are at least two identical pairs among this pair. This already means that the sequence is periodic.

We now choose among all such identical pairs two identical pairs with the smallest numbers. Let this pair with some numbers $(F_a, F_{a+1})$and (F_b, F_{b+1}). Let us prove that $a = 1$. Indeed, otherwise, for them there are previous pairs $(F_{a-1}, F_a)$and (F_{b-1}, F_b), which, by the property of Fibonacci numbers, will also be equal to each other. However, this contradicts the fact that we have chosen matching pairs with the smallest numbers, which was to be proved.

Literature

- Ronald Graham, Donald Knuth, Oren Patashnik. **Specific mathematics** [1998]

21 Комментариев **e-maxx** 1 **Войти**

Рекомендовать 7 Поделиться **Лучшее в начале**

Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ ИЛИ ЧЕРЕЗ DISQUS ?

Имя

Spellishment • 6 лет назад

А где, собственно, различные реализации алгоритма? В статье только формулы.

6 • Ответить • Поделиться ›

Guest → Spellishment • 6 лет назад

Да, мне тоже интересно...

6 • Ответить • Поделиться ›

Ivan Nikulin → Guest • 6 лет назад

f[0]=f[1]=1;
for (int i=2; i<=n; i++)
f[i]=f[i-1]+f[i-2];
Здесь не написано только это, насколько я вижу. Ссылка на возведение в степень дается, жадный алгоритм реализовать видимо не всем дано :) Чего вам еще не хватает?

P.S. Кстати, задача на перевод числа в Фиббоначиеву систему счисления: www.e-olimp.com/problems/1378/

8 • Ответить • Поделиться ›

nurda • 3 года назад

b[0]=0;
b[1]=1;
for(int i=2;i<=n;i++)
b[i]=b[i-1]+b[i-2];

2 • Ответить • Поделиться ›

Samandar Ravshanov • 6 лет назад

Кому интересно на питоне можно реализовать такими способами:

def fibonacci(**mnum**):
"фунция выводит всех чисел Фибоначчи до определенного"
a,b, fiblist = 0, 1, []

```
    while a < mnum:
        fiblist.append(a)
        a, b = b, a+b
    return fiblist
```

----

def fib(**n**):
"рекурсивная функция"

```
  if n < 3:
      return 1
  return fib(n-1) + fib(n-2)
```

2 ^ | v • Ответить • Поделиться ›

Louise ➦ Samandar Ravshanov • 6 лет назад

Второй вариант взорвётся уже при небольших n.

14 ^ | v • Ответить • Поделиться ›

Herman Yanush ➦ Samandar Ravshanov • 5 месяцев назад

Кому интересно, как перевести числа в фибоначчиеву систему:
https://bitbucket.org/snipp...

3 ^ | v • Ответить • Поделиться ›

Sonych KO ➦ Samandar Ravshanov • 4 года назад

или как-то так на генераторах

```
def fib():
a, b = 0, 1
while True:
yield a
a, b = b, a+b
```

^ | v • Ответить • Поделиться ›

Аслан Абисалов • 2 года назад

Период начинается с 0. Т.к. из пары соседних эл-то однозначно определен предыдущий, то первой пары нет
И 0 не более 1 раза внутри периода, т.к. после 0 A начинается послед 0 A A 2A итд (все по модулю P конечно)- в ней уже не встретится 0 1, если только не
A=1 или A=-1, в последнем случае 0 встречается 1 раз внутри периода
Вопрос - как узнать, встретится ли 0 -1?

^ | v • Ответить • Поделиться ›

Куаныш • 3 года назад

```
#include <iostream>

#include <cstdio>

#define ll long long

const ll mod = (ll)1e9+7;
```

using namespace std;

struct matrix

{

ll val[2][2],n,m,t[2][2];

void operator *=(matrix x)

{

t[0][0]=t[0][1]=0;

t[1][0]=t[1][1]=0;

for (int i=0;i<n;i++) for="" (int="" j="0;j<x.m;j++)" for="" (int="" u="0;u<m;u++)" t[i][j]="(val[i][u]*x.val[u][j]+t[i][j])%mod;" val[0][0]="t[0][0];" val[0][1]="t[0][1];" val[1][0]="t[1][0];" val[1][1]="t[1][1];" }="" }i,a,b,c;="" matrix="" bin_pow(matrix="" a,ll="" n)="" {="" if="" (n="=0)" return="" i;="" matrix="" b="bin_pow(a,n/2);" b*="b;" if="" (n%2!="0)" b*="a;" return="" b;="" }="" int="" main="" ()="" {="" ll="" n;="" scanf("%i64d",="" &n);="" a.val[0][0]="1;" a.val[0][1]="1;" a.n="1;" a.m="2;" i.n="2;" i.m="2;" b.n="2;" b.m="2;" b.val[0][0]="0;" b.val[0][1]="B.val[1][0]=B.val[1][1]=1;" i.val[0][0]="1;" i.val[0][1]="0;" i.val[1][0]="0;" i.val[1][1]="1;" b="bin_pow(B,N);" a*="B;" cout<<a.val[0][0];="" return="" 0;="" }="">

^ | v • Ответить • Поделиться ›

Куаныш ➦ Куаныш • 3 года назад

код

^ | v • Ответить • Поделиться ›

Куаныш ➦ Куаныш • 3 года назад

code

^ | v • Ответить • Поделиться ›

Куаныш ➦ Куаныш • 3 года назад

коде

^ | v • Ответить • Поделиться ›

нурда куренше • 3 года назад

long int n,b[1111],h,k=0;
cin>>n;
b[0]=0;
b[1]=1;
for(int i=2;i<=n;i++)
b[i]=b[i-1]+b[i-2];
h=b[n];
for(int i=1;i<=sqrt(h);i++)
{
if(h%i==0&&h/i!=i)k+=2;

else if(h%i==0)k++;
}
cout<<k; Еще="" до="" нашей="" эры,="" великий="" мудрец="" Фибоначчи="" ввёл="" в="" мир="" математики="" новую="" всеми="" известную="" последовательность="" Фиббоначи,="" где="" каждый="" элемент="" находится="" путём="" сложения="" двух="" предыдущих.="" Потом="" появилась="" великая="" задача="" Фибоначчи,="" где="" по="" заданному="" числу="" n,="" надо="" найти="" количество="" делителей="" числа="" f[n].="" Данная="" задача="" оставалось="" нерешённой="" до="" настоящего="" времени,="" и="" именно="" вам="" стоит="" решить="" эту="" задачу.="" --------------------="" входные="" данные="" 0="" выходные="" данные="" 0="" ---------------------="" входные="" данные="" 2="" выходные="" данные="">

^ | ⌄ • Ответить • Поделиться ›

Владислав • 5 лет назад

Хотелось бы реализацию логарифмического алгоритма.

^ | ⌄ • Ответить • Поделиться ›

thatskriptkid ➦ Владислав • 4 года назад

http://pastebin.com/53n4pwM6

Вот, посмотрите, можете проверить))

^ | ⌄ • Ответить • Поделиться ›

arcadiaq ➦ thatskriptkid • 4 года назад

У Вас линейная сложность, а не логарифмическая

^ | ⌄ • Ответить • Поделиться ›

Yura Akatov ➦ arcadiaq • 3 года назад

http://pastebin.com/sfhLWgJZ

^ | ⌄ • Ответить • Поделиться ›

Luka • 5 лет назад

f(n+1)*f(n-1) - f(n)*f(n) = (-1)^n
Википедия: f(n+1)*f(n+2) - f(n)*f(n+3) = (-1)^n
Неувязочка... (в статье формула не верна)

^ | ⌄ • Ответить • Поделиться ›

Кирилл ➦ Luka • 4 года назад

В статье все правильно написано, проверьте сами + на википедии