GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

Program for Fibonacci numbers

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 141, ……..

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

[Tex] F_n = F_{n-1} + F_{n-2}[/Tex]

with seed values

[Tex] F_0 = 0 \quad\text{and}\quad F_1 = 1. [/Tex]

Write a function *int fib(int n)* that returns [Tex]F_n[/Tex]. For example, if *n* = 0, then *fib()* should return 0. If n = 1, then it should return 1. For n > 1, it should return [Tex]F_{n-1} + F_{n-2}[/Tex]

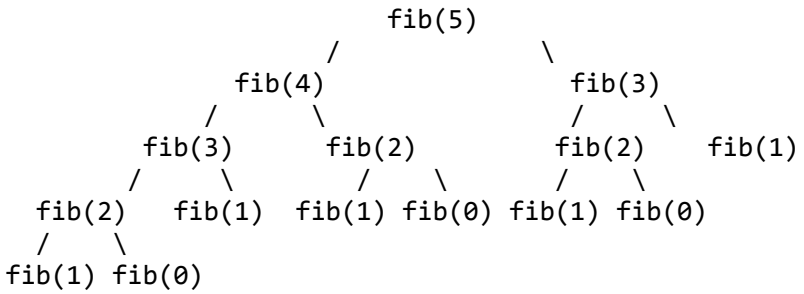
Following are different methods to get the nth Fibonacci number.

Method 1 (Use recursion)
A simple method that is a direct recusrive implementation mathematical recurance relation given above.

```
#include<stdio.h>
int fib(int n)
{
   if (n <= 1)
      return n;
   return fib(n-1) + fib(n-2);
}

int main ()
{
  int n = 9;
  printf("%d", fib(n));
  getchar();
  return 0;
}
```

Time Complexity: T(n) = T(n-1) + T(n-2) which is exponential.
We can observe that this implementation does a lot of repeated work (see the following recursion tree). So this is a bad implementation for nth Fibonacci number.



Extra Space: O(n) if we consider the fuinction call stack size, otherwise O(1).

Method 2 (Use Dynamic Programming)
We can avoid the repeated work done is the method 1 by storing the Fibonacci numbers calculated so far.

```
#include<stdio.h>

int fib(int n)
{
  /* Declare an array to store fibonacci numbers. */
  int f[n+1];
  int i;

  /* 0th and 1st number of the series are 0 and 1*/
  f[0] = 0;
  f[1] = 1;
```

```
  for (i = 2; i <= n; i++)
  {
      /* Add the previous 2 numbers in the series
         and store it */
      f[i] = f[i-1] + f[i-2];
  }

  return f[n];
}

int main ()
{
  int n = 9;
  printf("%d", fib(n));
  getchar();
  return 0;
}
```

Time Complexity: O(n)
Extra Space: O(n)

Method 3 (Space Otimized Method 2)
We can optimize the space used in method 2 by storing the previous two numbers only because that is all we need to get the next Fibannaci number in series.

```
#include<stdio.h>
int fib(int n)
{
  int a = 0, b = 1, c, i;
  if( n == 0)
    return a;
  for (i = 2; i <= n; i++)
  {
     c = a + b;
     a = b;
     b = c;
  }
  return b;
}

int main ()
{
  int n = 9;
  printf("%d", fib(n));
  getchar();
  return 0;
}
```

Time Complexity: O(n)
Extra Space: O(1)

Method 4 (Using power of the matrix {{1,1},{1,0}})

This another O(n) which relies on the fact that if we n times multiply the matrix M = {{1,1},{1,0}} to itself (in other words calculate power(M, n)), then we get the (n+1)th Fibonacci number as the element at row and column (0, 0) in the resultant matrix.

The matrix representation gives the following closed expression for the Fibonacci numbers:
[Tex]
\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}. [/Tex]

```
#include <stdio.h>

/* Helper function that multiplies 2 matricies F and M of size 2*2, and
  puts the multiplication result back to F[][] */
void multiply(int F[2][2], int M[2][2]);

/* Helper function that calculates F[][] raise to the power n and puts the
  result in F[][]
  Note that this function is desinged only for fib() and won't work as genera
  power function */
void power(int F[2][2], int n);

int fib(int n)
{
  int F[2][2] = {{1,1},{1,0}};
  if (n == 0)
      return 0;
  power(F, n-1);

  return F[0][0];
}

void multiply(int F[2][2], int M[2][2])
{
  int x =  F[0][0]*M[0][0] + F[0][1]*M[1][0];
  int y =  F[0][0]*M[0][1] + F[0][1]*M[1][1];
  int z =  F[1][0]*M[0][0] + F[1][1]*M[1][0];
  int w =  F[1][0]*M[0][1] + F[1][1]*M[1][1];

  F[0][0] = x;
  F[0][1] = y;
  F[1][0] = z;
  F[1][1] = w;
}

void power(int F[2][2], int n)
{
  int i;
  int M[2][2] = {{1,1},{1,0}};

  // n - 1 times multiply the matrix to {{1,0},{0,1}}
  for (i = 2; i <= n; i++)
      multiply(F, M);
}
```

```
/* Driver program to test above function */
int main()
{
  int n = 9;
  printf("%d", fib(n));
  getchar();
  return 0;
}
```

Time Complexity: O(n)
Extra Space: O(1)

Method 5 (Optimized Method 4)
The method 4 can be optimized to work in O(Logn) time complexity. We can do recursive multiplication to get power(M, n) in the prevous method (Similar to the optimization done in this post)

```
#include <stdio.h>

void multiply(int F[2][2], int M[2][2]);

void power(int F[2][2], int n);

/* function that returns nth Fibonacci number */
int fib(int n)
{
  int F[2][2] = {{1,1},{1,0}};
  if (n == 0)
    return 0;
  power(F, n-1);
  return F[0][0];
}

/* Optimized version of power() in method 4 */
void power(int F[2][2], int n)
{
  if( n == 0 || n == 1)
      return;
  int M[2][2] = {{1,1},{1,0}};

  power(F, n/2);
  multiply(F, F);

  if (n%2 != 0)
     multiply(F, M);
}

void multiply(int F[2][2], int M[2][2])
{
  int x =  F[0][0]*M[0][0] + F[0][1]*M[1][0];
  int y =  F[0][0]*M[0][1] + F[0][1]*M[1][1];
```

```
  int z =  F[1][0]*M[0][0] + F[1][1]*M[1][0];
  int w =  F[1][0]*M[0][1] + F[1][1]*M[1][1];

  F[0][0] = x;
  F[0][1] = y;
  F[1][0] = z;
  F[1][1] = w;
}

/* Driver program to test above function */
int main()
{
  int n = 9;
  printf("%d", fib(9));
  getchar();
  return 0;
}
```

Time Complexity: **O(Logn)**
Extra Space: O(Logn) if we consider the function call stack size, otherwise O(1).

Please write comments if you find the above codes/algorithms incorrect, or find other ways to solve the same problem.

References:
http://en.wikipedia.org/wiki/Fibonacci_number
http://www.ics.uci.edu/~eppstein/161/960109.html

Related Topics:

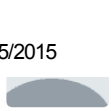
- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: Dynamic Programming, MathematicalAlgo

Tweet 7 +1 10

Writing code in comment? Please use **ideone.com** and share the link here.

83 Comments GeeksforGeeks 1 Login
Recommend 3 Share Sort by Newest
Join the discussion…



Shyam choudhary • 2 months ago

here my implementation of the same using dynamic programming in C
http://shyamcodersphere.blogsp...

• Reply • Share ›

Pramod Setlur • 2 months ago

```
import java.util.Hashtable;

public class Fibonacci

{

static Hashtable<integer, integer=""> hashMapRecursive = new Hashtable<integer, integer="">
();

static Hashtable<integer, integer=""> hashMapIterative = new Hashtable<integer, integer="">();

public static void main(String[] args)

{

int n = 40 ;

long startTime = System.currentTimeMillis();

long answer = recursiveFib(n);
```

see more

• Reply • Share ›

shubham vashishtha • 3 months ago

thank you!

• Reply • Share ›

Deepender Singla • 4 months ago

I think in 3rd solution Method 3 (Space Otimized Method 2), a need to be initialised as 1.

• Reply • Share ›

Farhad Vanaei • 4 months ago

Awesome Explanation.

• Reply • Share ›

person464 • 6 months ago

you have a mistake... 141 should be 144

in the sequence:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 141
5 ^ | v • Reply • Share ›

sourabh_hhh • 6 months ago
For better understanding of Method 5 :

See Example 1.19 in SCIP's books.
1) https://mitpress.mit.edu/sicp/...
A video tutorial by MathDoctorBob

2) https://www.youtube.com/watch?...
3 ^ | v • Reply • Share ›

raviteja1452 • 6 months ago
Super Explanation :D
^ | v • Reply • Share ›

Johana Halili • 7 months ago
in sml i mean
^ | v • Reply • Share ›

Johana Halili • 7 months ago
how do i create a function that proves if a number is a fibonaci number of not? fib=fn:int->bool
^ | v • Reply • Share ›

Vikas Malviya ➚ Johana Halili • 7 months ago
if n is a fibonacci number then, 5(n^2)+4 or 5(n^2)-4 are perfect squares
1 ^ | v • Reply • Share ›

sadiquee • 7 months ago
excellent Mr.vaibhav Joshi
^ | v • Reply • Share ›

Jon Snow • 7 months ago
From exponential to logarithmic. what a great reduction of both time and space complexity!!!
1 ^ | v • Reply • Share ›

Siva Krishna • 8 months ago
We can also do in O(logn) time by finding [pow(φ, n)/sqrt(5)]
^ | v • Reply • Share ›

mauricepatel37 ➚ Siva Krishna • 6 months ago

@Siva Krishna SQRT function is generally
avoided since you don't know what method the inbuilt function is using and how much is its complexity.
So it is better to check (i*i==t) instead of (i==sqrt(t))!

1 • Reply • Share ›

Guest Siva Krishna • 6 months ago

@Siva Krishna in most of the cases SQRT function is avoided since you don't know what method the inbuilt function is using and how much is its complexity.
So it is better to check (i*i==t) instead of (i==sqrt(t))!

• Reply • Share ›

Vãîbhåv Joshî • 10 months ago

above the 12th no of Fibonacci.....is 144 instead of 141

3 • Reply • Share ›

10 • 10 months ago

Question->http://www.spoj.com/problems/F... Basically question is to find the sum of fibonacci numbers starting from N to M I did this question using the factor (1+sqrt(5))/2.I founded N,N+1 using this factor by O(log N) multiplication.Then finally i traversed all the numbers >from N+2 to M and side by side adding their sum.My code takes O(N-M+logN) time.But i am getting time limit exceeded on this solution.What can i do to improve the time complexity of my solution.I thought of doing using dynamic programming will it be faster ??plz help

```
#include<iostream>
#include<math.h>
long int power_multiply(long int,long int&);
using namespace std;
int main()
{
long int N,M,a,b,sum=0,c,t;
cin>>t;
while(t--){
cin>>N>>M;
a=power_multiply(N,b);//i passed b as reference so that a=f(n),b=f(n+1) are simultaneously
```

see more

• Reply • Share ›

mauricepatel37 10 • 6 months ago

you are getting TLE because of using SQRT function. in most of the cases SQRT function is avoided since you don't know what method the inbuilt function is using and how much is its complexity.
it is better to check (i*i==t) instead of (i==sqrt(t))!

So that is the reason we are using one of the five method described in the article

• Reply • Share ›

krishna 10 • 9 months ago

take a look
http://www.physicsforums.com/s...

• Reply • Share ›

Programmer • 10 months ago

We also reduce multiplication..

```
int x = F[0][0]*M[0][0] + F[0][1]*M[1][0];
int y = F[0][0]*M[0][1] + F[0][1]*M[1][1];
int z = F[1][0]*M[0][0] + F[1][1]*M[1][0];
int w = F[1][0]*M[0][1] + F[1][1]*M[1][1];
```

for this...

```
int z = F[0][0];
int w = F[0][1];
int x = F[0][0]*M[0][0] + F[0][1]*M[1][0];
int y = F[0][0]*M[0][1] + F[0][1]*M[1][1];
```

Bcz Upper row is copy of lower low in next multiplication
For mathod :- 3

2 • Reply • Share ›

Wei Xue • a year ago

The time complexity of dynamic programming is NOT O(n). It looks O(n), but actually is O(n^2). When n is getting larger, the operation of addition will increase, which is not O(1).

3 • Reply • Share ›

mauricepatel37 Wei Xue • 6 months ago

@Wei Xue i don't think so.even if n is very very large then also for one iteration of loop
1)accessing two elements of array in O(1) time
2)addition of two numbers O(1) time.
So how would operation of addition increase?

• Reply • Share ›

Wei Xue mauricepatel37 • 6 months ago

addition is not O(1), the time for addition will get get longer when N is bigger. In CLRS 3rd, "but we shall assume that each execution of the ith line takes time ci, where ci is a constant. " It does apply here. How would you add two 4-bit numbers? (if use 32 bits to store them, other 28 bit will be 0). How would you add two 32-bit numbers? The cost of adding 32-bit numbers will be greater than adding 4-bit numbers. Therefore, if F(n) is extremely large, the addition will cost

you a lot of time~

• Reply • Share ›

Aditya Goel ➚ Wei Xue • 3 months ago

How much time we are taking about? Suppose addition of two big numbers takes time c, c it is still very small compared to n. The computer doesn't scan anything; it has dedicated hardware that performs integer addition in a single cycle irrespective of number is small or large.
For your argument sake I can say overall time complexity is O(cn), but not in any case O(n^2).

• Reply • Share ›

Alejo Hausner ➚ Aditya Goel • a month ago

For large n, F(n) won't fit in a 32-bit int. Hence, you will have to store the digits of F(n) in an array, and you will switch to writing special code to do multiplications an additions "by hand". This is what the BigInteger class in Java does.

As you may remember from back in elementary school when you learned to do addition digit-by-digit, the amount of work needed to add two numbers is linear in the number of digits involved. A multiplication takes work which is quadratic in the number of digits.

• Reply • Share ›

Pegasi • a year ago

The 12th fib number is 144 apparently

• Reply • Share ›

samtron92 ➚ Pegasi • a year ago

true,, 141 is incorrect it should be 144

• Reply • Share ›

Guest ➚ Pegasi • a year ago

true, 141 is incorrect

• Reply • Share ›

s.a. • a year ago

can the last method be used to print the fibonacci series i.e. in (log n) time ? if so, how?

2 • Reply • Share ›

Pegasi ➚ s.a. • a year ago

Yes the recursion does it in O(log n) time. Here is an iterative power function:

void power(int F[2][2], int n)

```
{
int P[2][2] = {{1,1},{1,0}};
if (n % 2 != 0)
multiply(P, F);
while (n > 0)
{
n /= 2;
multiply(F, F);
}
}
```

1 • Reply • Share ›

s.a. • a year ago

can the last method be used to print the fibonacci series i.e. in (nlog n) time ? if so, how?

• Reply • Share ›

J Reyes • a year ago

In Java

```
public static long fib(int n) {
if(n == 0) { return n; }
n = n - 1;

BigDecimal eigen1 = new BigDecimal("-.61803398875");
BigDecimal eigen2 = new BigDecimal("1.61803398875");
BigDecimal det = new BigDecimal("-.4472135955" );
BigDecimal fib = eigen1.pow(n+1).subtract(eigen2.pow(n+1)).multiply(det);

return fib.longValue();
}
```

• Reply • Share ›

A Friend from hiddle leaf • a year ago

comment section sucks cant write while loop

• Reply • Share ›

A Friend from hiddle leaf • a year ago

Comment section sucks

concept-

```
int t0=1;
it t1=1;
cout<<t0<<endl; while(t1<20000)="" {="" cout<<t1<<endl;="" int="" temp="t1;" t1="to+t1;"
t0="t1;" }="">
```

• Reply • Share ›

A Friend from hiddle leaf • a year ago

```
//Take this losers, will run directly in Dev C++ without any change
//THIS-IS-NOOB

#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
int maxRange;
cout<<"Hi there!">>endl;
cout<<"Enter the maximum range for fibonacci series:";
cin>>maxRange;
int to = 1;
int t1= 1;
cout<<t0<<endl; while(t1="" <="" maxrange)="" {="" cout<<t1<<endl;="" int="" temp="t1;"
t1="t0" +="" t1;="" t0="temp;" }="" getch();="" return="" 0;="" }="">
```

▲ | ▼ • Reply • Share ›

A Friend from hiddle leaf • a year ago

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
int maxRange;
cout<<"Hi there!"<<endl; cout<<"enter="" the="" maximum="" range="" for="" fibonaaci=""
series:";="" cin="">>maxRange;
int t0 = 1;
int t1 = 1;
cout<<t0<<endl; while(t1="" <="" maxrange)="" {="" cout<<t1<<endl;="" int="" temp="t1;"
t1="t0" +="" t1;="" t0="temp;" }="" getch();="" return="" 0;="" }="" there="" was="" a=""
problem="" with="" website="" text="" formatting="">
```

▲ | ▼ • Reply • Share ›

A Friend from hiddle leaf • a year ago

```
//Take this losers, will run without any change in Dev C++
//THIS-IS-NOOB
#include<iostream>

#include<conio.h>

using namespace std;
```

```
int main()

{

int maxRange;

cout<<"Hi there!"<<endl; cout<<"enter="" the="" maximum="" range="" for="" fibonaaci="" series:";="" cin="">>maxRange;

int t0 = 1;

int t1 = 1;

cout<<t0<<endl; while(t1="" <="" maxrange)="" {="" cout<<t1<<endl;="" int="" temp="t1;" t1="t0" +="" t1;="" t0="temp;" }="" getch();="" return="" 0;="" }="">
```

• Reply • Share ›

Karim • 2 years ago

well there is another way, is to find where do we use fibonnaci, in the golden number. so here is the code.
phi=(1+sqrt(5))/2
fib(n) = floor(phi^n/sqrt(5) +1/2)

2 • Reply • Share ›

Sidhant ➚ Karim • a year ago

Phi is irrational...so while coding u cant get it accurate enough..due to which you cant generate Fibonacci numbers accurately

1 • Reply • Share ›

aditya • 2 years ago

plz sum1 post the solution of making program of fibonacci series using golden ratio and it should not be recrsive.

1 • Reply • Share ›

Ronny • 2 years ago

@GeeksforGeeks

In method 4 statement and its description the fibonacci matrix is expressed as {{1,1},{0,1}}, whereas in the program follwing the description and the method 5 uses fibonacci matrix as {{1,1},{1,0}}.

there is a typo
kindly update the post

• Reply • Share ›

GeeksforGeeks ➚ Ronny • 2 years ago
Thanks for pointing this out. We have corrected the typo.
• Reply • Share ›

Ronny ➚ GeeksforGeeks • 2 years ago
@GeeksforGeeks

There is still a typo in the description of the method 4.(only header has been corrected, description still needs to be corrected)
• Reply • Share ›

GeeksforGeeks ➚ Ronny • 2 years ago
Thanks Ronny, we have corrected it now.
• Reply • Share ›

Kalyani Arla • 2 years ago
if the callee function is above the caller function, you need not declare it(callee)
• Reply • Share ›

Mohammad Faizan Ali • 2 years ago
5th solution is awesome.
keep up the very good work.
• Reply • Share ›

Atiq Butt • 2 years ago
0 0
1 1
2 10
3 101
4 10110
5 10110101
so on
2 • Reply • Share ›

Atiq Butt • 2 years ago
I need fibinoci of bit string like 0 for 0 1 for 1 but for 2 it must be 10 and for 3 it must be 101 and so on.
• Reply • Share ›

Hardik Hadvani • 2 years ago
Hey Adminr,

Excellent article for the Fibonacci series of course this blog is doing a very good job of serving useful information. I&#039m proud to be a part of its Readers community.

For the Fibonacci programs in different languange like C language, JAVA, C# must visit http://www.hhhprogram.com/2013....

• Reply • Share ›

Load more comments

Subscribe Add Disqus to your site Privacy

- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching

- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 3 hours ago

@geeksforgeeks, Some rights reserved Contact Us!
Powered by WordPress & MooTools, customized by geeksforgeeks team