Sum of Fibonacci Numbers in a range

Difficulty Level : Medium • Last Updated : 17 Apr, 2019

Given a range **[l, r]**, the task is to find the sum **fib(l) + fib(l + 1) + fib(l + 2) + ….. + fib(r)** where **fib(n)** is the **n**th Fibonacci number.

Related Articles

> ***Output:*** *11*
>
> *fib(2) + fib(3) + fib(4) + fib(5) = 1 + 2 + 3 + 5 = 11*
>
> ***Input:*** *l = 4, r = 8*
>
> ***Output:*** *50*

Recommended: Please try your approach on ***{IDE}*** first, before moving on to the solution.

Naive approach: Simply calculate **fib(l) + fib(l + 1) + fib(l + 2) + ….. + fib(r)** in **O(r – l)** time complexity.
In order to find **fib(n)** in **O(1)** we will take help of Golden Ratio.
Fibonacci calculation using Binet's Formula

> *fib(n) = phi^n – psi^n) / ?5*
>
> *Where,*
>
> *phi = (1 + sqrt(5)) / 2 which is roughly equal to 1.61803398875*
>
> *psi = 1 – phi = (1 – sqrt(5)) / 2 which is roughly equal to 0.61803398875*

Below is the implementation of the above approach:

C++

```
// C++ implementation of the approach
#include <bits/stdc++.h>
using namespace std;
#define ll long long int

// Function to return the nth Fibonacci number
int fib(int n)
{
    double phi = (1 + sqrt(5)) / 2;
    return round(pow(phi, n) / sqrt(5));
}

// Function to return the required sum
ll calculateSum(int l, int r)
{

    // To store the sum
    ll sum = 0;
    for (int i = l; i <= r; i++)
        sum += fib(i);

    return sum;
}

// Driver code
int main()
{
    int l = 4, r = 8;
    cout << calculateSum(l, r);

    return 0;
}
```

Java

```
// Java implementation of the approach
import java.lang.Math;
class GFG
{

// Function to return the nth Fibonacci number
static int fib(int n)
{
    double phi = (1 + Math.sqrt(5)) / 2;
    return (int)Math.round(Math.pow(phi, n) / Math.sqrt(5));
}

// Function to return the required sum
static int calculateSum(int l, int r)
{

    // To store the sum
    int sum = 0;

    // Calculate the sum
    for (int i = l; i <= r; i++)
        sum += fib(i);

    return sum;
}

// Driver code
public static void main(String[] args)
{
    int l = 4, r = 8;
    System.out.println(calculateSum(l, r));
```

```
//This code is contributed by Code_Mech.
```

Python3

```
# Python3 implementation of the approach

# Function to return the nth
# Fibonacci number
def fib(n):
    phi = ((1 + (5 ** (1 / 2))) / 2);
    return round((phi ** n) / (5 ** (1 / 2)));

# Function to return the required sum
def calculateSum(l, r):

    # To store the sum
    sum = 0;

    # Calculate the sum
    for i in range(l, r + 1):
        sum += fib(i);

    return sum;

# Driver Code
if __name__ == '__main__':
    l, r = 4, 8;
    print(calculateSum(l, r));

# This code contributed by Rajput-Ji
```

C#

```
// C# implemenatation of above approach
using System;

class GFG
{

// Function to return the nth Fibonacci number
static int fib(int n)
{
    double phi = (1 + Math.Sqrt(5)) / 2;
    return (int)Math.Round(Math.Pow(phi, n) / Math.Sqrt(5));
}
```

```
    // To store the sum
    int sum = 0;

    // Calculate the sum
    for (int i = l; i <= r; i++)
        sum += fib(i);

    return sum;
}

// Driver code
public static void Main()
{
    int l = 4, r = 8;
    Console.WriteLine(calculateSum(l, r));
}
}

/* This code contributed by PrinciRaj1992 */
```

PHP

```
<?php
// PHP implementation of the approach

// Function to return the nth
// Fibonacci number
function fib($n)
{
    $phi = (1 + sqrt(5)) / 2;
    return (int)round(pow($phi, $n) / sqrt(5));
}

// Function to return the required sum
function calculateSum($l, $r)
{

    // To store the sum
    $sum = 0;

    // Calculate the sum
    for ($i = $l; $i <= $r; $i++)
        $sum += fib($i);

    return $sum;
}
```

```
echo calculateSum($l, $r);

// This code is contributed by mits
?>
```

Output:

```
50
```

Efficient approach: The idea is to find the relationship between the sum of Fibonacci numbers and n^{th} Fibonacci number and use **Binet's Formula** to calculate its value.

Relationship Deduction

1. F(i) refers to the i^{th} Fibonacci number.
2. S(i) refers to sum of Fibonacci numbers till F(i).

> *We can rewrite the relation $F(n + 1) = F(n) + F(n - 1)$ as below:*
> $F(n - 1) = F(n + 1) - F(n)$
> *Similarly,*
> $F(n - 2) = F(n) - F(n - 1)$
> ...
> ...
> ...
> $F(0) = F(2) - F(1)$
>
> *Adding all the equations, on left side, we have*
> $F(0) + F(1) + ... + F(n - 1)$ *which is* $S(n - 1)$

Therefore,
$S(n - 1) = F(n + 1) - F(1)$
$S(n - 1) = F(n + 1) - 1$
$\mathbf{S(n) = F(n + 2) - 1}$

In order to find **S(n)**, simply calculate the **$(n + 2)^{th}$** Fibonacci number and subtract **1** from the result.

Therefore

$S(l, r) = F(r + 2) - 1 - (F(l + 1) - 1)$

$\mathbf{S(l, r) = F(r + 2) - F(l + 1)}$

C++

```
// C++ implementation of the approach
#include <bits/stdc++.h>
using namespace std;

// Function to return the nth Fibonacci number
int fib(int n)
{
    double phi = (1 + sqrt(5)) / 2;
    return round(pow(phi, n) / sqrt(5));
}

// Function to return the required sum
int calculateSum(int l, int r)
{

    // Using our deduced result
    int sum = fib(r + 2) - fib(l + 1);
    return sum;
}

// Driver code
int main()
{
    int l = 4, r = 8;
    cout << calculateSum(l, r);

    return 0;
}
```

Java

```
// Java implementation of the approach
class GFG
{

// Function to return the nth Fibonacci number
static int fib(int n)
{
    double phi = (1 + Math.sqrt(5)) / 2;
    return (int) Math.round(Math.pow(phi, n) / Math.sqrt(5));
}
```

```
    // Using our deduced result
    int sum = fib(r + 2) - fib(l + 1);
    return sum;
}

// Driver code
public static void main(String[] args)
{
    int l = 4, r = 8;
    System.out.println(calculateSum(l, r));

}
}

// This code is contributed by 29AjayKumar
```

Python3

```
# Python3 implementation of the approach
import math

# Function to return the nth
# Fibonacci number
def fib(n):

    phi = (1 + math.sqrt(5)) / 2;
    return int(round(pow(phi, n) /
                         math.sqrt(5)));

# Function to return the required sum
def calculateSum(l, r):

    # Using our deduced result
    sum = fib(r + 2) - fib(l + 1);
    return sum;

# Driver code
l = 4;
r = 8;
print(calculateSum(l, r));

# This code is contributed by mits
```

C#

```
// C# implementation of the approach
```

```
{

// Function to return the nth Fibonacci number
static int fib(int n)
{
    double phi = (1 + Math.Sqrt(5)) / 2;
    return (int) Math.Round(Math.Pow(phi, n) /
                            Math.Sqrt(5));
}

// Function to return the required sum
static int calculateSum(int l, int r)
{

    // Using our deduced result
    int sum = fib(r + 2) - fib(l + 1);
    return sum;
}

// Driver code
public static void Main()
{
    int l = 4, r = 8;
    Console.WriteLine(calculateSum(l, r));
}
}

// This code is contributed
// by Akanksha Rai
```

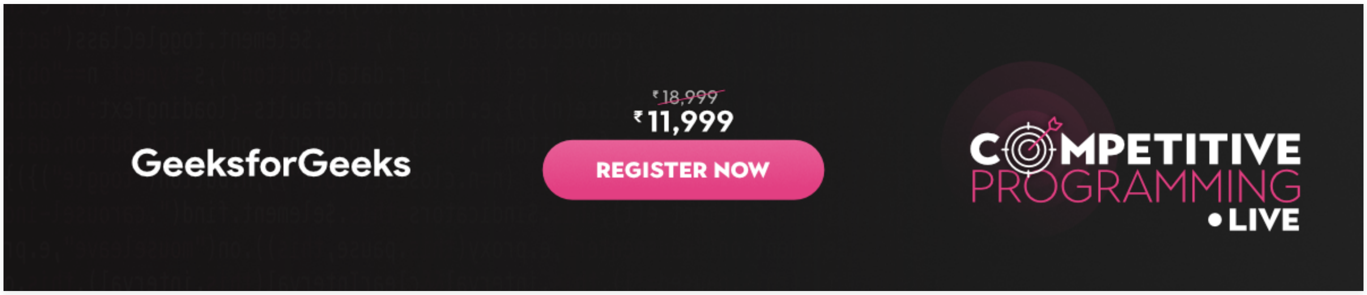
PHP

```
<?php
// PHP implementation of the approach

// Function to return the nth Fibonacci number
function fib($n)
{
    $phi = (1 + sqrt(5)) / 2;
    return (int) round(pow($phi, $n) / sqrt(5));
}

// Function to return the required sum
function calculateSum($l, $r)
{

    // Using our deduced result
    $sum = fib($r + 2) - fib($l + 1);
```

```php
// Driver code
$l = 4; $r = 8;
echo(calculateSum($l, $r));

// This code is contributed by Code_Mech
?>
```

Output:

```
50
```



RECOMMENDED ARTICLES

Page : 1 2 3

not
09, Apr 20

02 **Sum of all Non-Fibonacci numbers in a range for Q queries**
27, Mar 20

03 **Numbers with a Fibonacci difference between Sum of digits at even and odd positions in a given range**
09, Apr 20

04 **Count numbers divisible by K in a range with Fibonacci digit sum for Q queries**
09, Apr 20

05, May 20

06 **Check if a M-th fibonacci number divides N-th fibonacci number**
11, Jul 18

07 **Count Fibonacci numbers in given range in O(Log n) time and O(1) space**
15, Jan 16

08 **Array range queries to count the number of Fibonacci numbers with updates**
19, Apr 20

♡ **Like** 0

|< Previous

Next >|

Article Contributed By :

ShivamChauhan5
@ShivamChauhan5

Vote for difficulty

Easy Normal Medium Hard Expert

Improved By : Mithun Kumar, Code_Mech, princiraj1992, 29AjayKumar, Akanksha_Rai, Rajput-Ji

Article Tags : Fibonacci , Algorithms , Competitive Programming , Mathematical

Practice Tags : Mathematical, Fibonacci, Algorithms

Improve Article Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

0 Comments **GeeksforGeeks** **Disqus' Privacy Policy** 1 **Login**

Recommend Tweet Share **Sort by Newest**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

Subscribe **Add Disqus to your siteAdd DisqusAdd** **Do Not Sell My Data**

GeeksforGeeks