

AVAILABLE
LESSONS:*Lesson 1*
Iterations*Lesson 2*
Arrays*Lesson 3*
Time Complexity*Lesson 4*
Counting
Elements*Lesson 5*
Prefix Sums*Lesson 6*
Sorting*Lesson 7*
Stacks and
Queues*Lesson 8*
Leader*Lesson 9*
Maximum slice
problem*Lesson 10*
Prime and
composite
numbers*Lesson 11*

RESPECTABLE

FibFrog

START

Count the minimum number of jumps required for a frog to get to the other side of a river.

Programming language: C++ ▼

The Fibonacci sequence is defined using the following recursive formula:

$$F(0) = 0$$

$$F(1) = 1$$

$$F(M) = F(M - 1) + F(M - 2) \text{ if } M \geq 2$$

A small frog wants to get to the other side of a river. The frog is initially located at one bank of the river (position -1) and wants to get to the other bank (position N). The frog can jump over any distance $F(K)$, where $F(K)$ is the K -th Fibonacci number. Luckily, there are many leaves on the river, and the frog can jump between the leaves, but only in the direction of the bank at position N .

The leaves on the river are represented in a zero-indexed array A consisting of N integers. Consecutive elements of array A represent consecutive positions from 0 to $N - 1$ on the river. Array A contains only 0 s and/or 1 s:

- 0 represents a position without a leaf;
- 1 represents a position containing a leaf.

The goal is to count the minimum number of jumps in which the frog can get to the other side of the river (from position -1 to position N). The frog can jump between positions -1 and N (the banks of the river) and every position containing a leaf.

For example, consider array A such that:

$$A[0] = 0$$

$$A[1] = 0$$

$$A[2] = 0$$

$$A[3] = 1$$

$$A[4] = 1$$

$$A[5] = 0$$

Sieve of
Eratosthenes

Lesson 12

Euclidean
algorithm

Lesson 13

**Fibonacci
numbers**

Lesson 14

Binary search
algorithm

Lesson 15

Caterpillar
method

Lesson 16

Greedy
algorithms

Lesson 17

Dynamic
programming

Lesson 90

Tasks from
Indeed Prime
2015 challenge

Lesson 91

Tasks from
Indeed Prime
2016 challenge

Lesson 92

Tasks from
Indeed Prime
2016 College
Coders
challenge

Lesson 99

```
A[6] = 1
A[7] = 0
A[8] = 0
A[9] = 0
A[10] = 0
```

The frog can make three jumps of length $F(5) = 5$, $F(3) = 2$ and $F(5) = 5$.

Write a function:

```
int solution(vector<int> &A);
```

that, given a zero-indexed array A consisting of N integers, returns the minimum number of jumps by which the frog can get to the other side of the river. If the frog cannot reach the other side of the river, the function should return -1 .

For example, given:

```
A[0] = 0
A[1] = 0
A[2] = 0
A[3] = 1
A[4] = 1
A[5] = 0
A[6] = 1
A[7] = 0
A[8] = 0
A[9] = 0
A[10] = 0
```

the function should return 3, as explained above.

Assume that:

- N is an integer within the range $[0..100,000]$;
- each element of array A is an integer that can have one of the following values: 0, 1.

Complexity:

- expected worst-case time complexity is $O(N \cdot \log(N))$;
- expected worst-case space complexity is $O(N)$, beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2016 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

**For
programmers**

Lessons Challenges
Terms FAQ

For companies

Tour About
Pricing us
Blog Jobs
Privacy Terms
Cookies
API

**Sign up for our
newsletter:**

Information about
upcoming challenges,
solutions and lessons
directly in your inbox.

Social:

[f](#) [t](#) [in](#)

Contact us:

For customer support
queries:

UK +44 (0) 208 970
78 68

US 1-415-466-8085
support@codility.com

For sales queries:

UK +44 (0) 208 970
78 67

US 1-415-466-8085
sales@codility.com

© 2009–2016 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID
GB981191408. Registered office: 107 Cheapside, London EC2V 6DN