

Get account with free plan

See how Codility works from recruiter's point of view.

closed

Demo ticket

Session ID: demoBWA3CN-E4P
Time limit: 120 min.

Status: closed
Started on: 2014-01-16 00:40 UTC

Score:

100
of 100

★ 1. FrogJump
Count minimal number of jumps from position X to Y.

score: 100 of 100

Task description

A small frog wants to get to the other side of the road. The frog is currently located at position X and wants to get to a position greater than or equal to Y. The small frog always jumps a fixed distance, D.
Count the minimal number of jumps that the small frog must perform to reach its target.
Write a function:

```
class Solution { public int solution(int X, int Y, int D); }
```

that, given three integers X, Y and D, returns the minimal number of jumps from position X to a position equal to or greater than Y.
For example, given:

```
X = 10  
Y = 85  
D = 30
```

the function should return 3, because the frog will be positioned as follows:

- after the first jump, at position 10 + 30 = 40
- after the second jump, at position 10 + 30 + 30 = 70
- after the third jump, at position 10 + 30 + 30 + 30 = 100

Assume that:

- X, Y and D are integers within the range [1..1,000,000,000];
- X ≤ Y.

Complexity:

- expected worst-case time complexity is O(1);
- expected worst-case space complexity is O(1).

Copyright 2009–2014 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

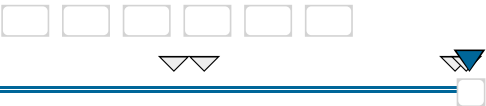
Programming language used: Java

Total time used: 12 minutes

Effective time used: 12 minutes

Notes: correct functionality and scalability

Task timeline



00:40:46 00:52:12

Code: 00:52:11 UTC, java, final, score: 100.00

```
01. // you can also use imports, for  
    // example:  
02. // import java.math.*;  
03. class Solution {  
04.     public int solution(int X, int Y,  
        int D) {  
05.         int distance = Y - X;  
06.         int rest = distance % D;  
07.         int hops = distance / D;  
08.  
09.         if(rest > 0){  
10.             return ++hops;  
11.         }  
12.         return hops;  
13.     }  
14. }
```

Analysis

Detected time complexity:
O(1)

test	time	result
example example test	0.290 s.	OK
simple1 simple test	0.300 s.	OK
simple2	0.290 s.	OK
extreme_position		

<div>Get account</div>	no jump needed	0.310 s.	OK
	small_extreme_jump one big jump	0.300 s.	OK
	many_jump1 many jumps, D = 2	0.310 s.	OK
	many_jump2 many jumps, D = 99	0.300 s.	OK
	many_jump3 many jumps, D = 1283	0.290 s.	OK
	big_extreme_jump maximal number of jumps	0.300 s.	OK
	small_jumps many small jumps	0.290 s.	OK