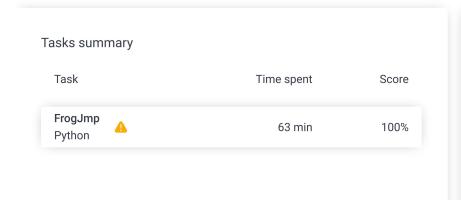
Codility_

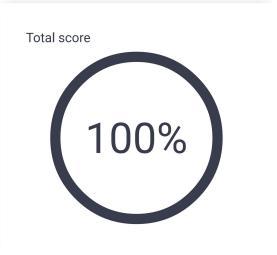
Candidate Report: training9DRVBH-TTR

Test Name:

Summary Timeline

Check out Codility training tasks





Tasks Details

1. FrogJmp
Count minimal number of Task Score Correctness Performance
jumps from position X to
Y.

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:

def solution(X, Y, 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

Solution

Programming language used: Python

Total time used: 63 minutes

Effective time used: 63 minutes

Notes: not defined yet

Task timeline

04:28:56 05:31:46

Code: 05:31:46 UTC, py, final, score: **100**

show code in pop-up

1 # you can write to stdout for debugging purposes, 2 # print("this is a debug message")

3

<釐清問題>

4

after the third jump, at position 10 + 30 + 30 + 30
 = 100

Write an efficient algorithm for the following assumptions:

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

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
Test results - Codility
```

```
計算從起始座標(X=10)抵達某一座標(Y=85)至少需要多少
6
        e.g.
7
    #
           X = 10 (起始座標)
            Y = 85 (須越過的target)
8
    #
9
    #
            D = 30 (每次跳躍的distance)
10
    #
            最終位置 = 10+(30*3)= 100 > 85(Y的位置),
11
            表示X需要跳3次才能跳過Y, 故return3
    #
12
13
14
15
    # <定義知識&製作解決方案>
        用三個變數做算術運算,並在X未移後的最終位置>Y的位置
16
17
    # <驗證&改進>
18
19
        可能的測資:
    #
20
           - (1,1,1)
21
    #
            - (10, 1000000000, 11) -> TIMEOUT ERROR (K
            - (1,1000000000,1)
22
23
    #
            - (31, 44648891, 13) # print(31+13*3434528
24
        需減少運算時間:
25
26
27
28
    #
        其他限制:
            X, Y and D are integers within the range [
29
    #
            X ≤ Y (X只會右移動或不移動)
30
31
32
    def solution(X, Y, D):
        return countMinNumOfJumpsFast(X,Y,D)
33
34
    def countMinNumOfJumps(start:int, target:int, dist
35
36
        final_position = start
        cnt_jumps = 0
37
        while final_position<target:
38
39
            final_position += distance
40
            cnt_jumps += 1
41
        return cnt_jumps
42
    def countMinNumOfJumpsFast(start:int, target:int,
43
44
           起始位置+ (位移*位移次數)= 最終位置 > Y的位置
45
        #
46
        # - 減少迴圈運算改善效能
47
48
        tmp = target - start
49
50
        jumps = int( tmp / distance)
        if tmp%distance != 0:
                                        # 若整除則終黑
51
52
            jumps += 1
53
54
        return jumps
55
56
    # When using:
57
         jumps = round(tmp/distance + 0.5)
58
59
    # Get the wrong answer:
    # Your test case: [1, 100000000, 1]
60
    # Returned value: 1000000000
61
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(1)



1.	0.036 s OK	
colla	pse all Correc	tness tests
▼	simple1 simple test	√ OK
1.	0.036 s OK	
2.	0.036 s OK	
▼	simp l e2	√ OK
1.	0.036 s OK	
2.	0.036 s OK	
V	extreme_position no jump needed	√ OK
1.	0.036 s OK	
2.	0.036 s OK	
•	small_extreme_jump one big jump	√ OK
1.	0.036 s OK	
colla	pse all Perfor n	nance tests
•	many_jump1 many jumps, D = 2	√ OK
1.	0.036 s OK	
•	many_jump2 many jumps, D = 99	√ OK
1.	0.036 s OK	
•	many_jump3 many jumps, D = 1283	√ OK
1.	0.036 s OK	
•	big_extreme_jump maximal number of jumps	√ OK
1.	0.036 s OK	
•	small_jumps many small jumps	√ 0K

The PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.