



ryan  
Other

PDF generated at: 28 Jul 2024 18:09:54 UTC

[View this report on HackerRank](#)

Score

100% • 80 / 80  
scored in CodePath TIP101: Unit 7 Assessment, Version A - Summer 2024 in 18 min 23 sec on 28 Jul 2024 10:49:43 PDT

Candidate Information

Email	concepting@protonmail.com
Test	CodePath TIP101: Unit 7 Assessment, Version A - Summer 2024
Candidate Packet	<a href="#">View</a>
Taken on	28 Jul 2024 10:49:43 PDT
Time taken	18 min 23 sec/ 60 min
Work Experience	< 1 years
Invited by	CodePath

Suspicious Activity detected

Code similarity

 Code similarity  
3 questions

Skill Distribution



There is no associated skills data that can be shown for this assessment

Tags Distribution



There is no associated tags data that can be shown for this assessment

Questions

Status	No.	Question	Time Taken	Skill	Score
	1	Mystery Time Complexity Multiple Choice	20 sec	-	5/5
	2	Recursive Prediction Multiple Choice	1 min 34 sec	-	5/5

✓	3	Base Case Scenario Multiple Choice	2 sec	-	5/5
✓	4	Predict the Recursive Call Multiple Choice	1 min 47 sec	-	5/5
✓	5	String Search Coding	5 min 2 sec	-	20/20 🚩
✓	6	Power Function Coding	5 min 47 sec	-	20/20 🚩
✓	7	Find the Domain Coding	3 min 30 sec	-	20/20 🚩

## 1. Mystery Time Complexity

✓ Correct

Multiple Choice

### Question description

What is the time complexity of the following function?

`lst` is a sorted list of integers. `target` is an integer. Assume `n` represents the length of `lst`.

```
def mystery_function(lst, target):  
    left, right = 0, len(lst) - 1  
  
    while left <= right:  
        mid = (left + right) // 2  
        mid_value = lst[mid]  
  
        if mid_value == target:
```

```
        return mid
    elif mid_value < target:
        left = mid + 1
    else:
        right = mid - 1

return -1
```

### Candidate's Solution

**Options:** (Expected answer indicated with a tick)

☐  $O(1)$

☒  $O(\log(n))$



☐  $O(n)$

☐  $O(n^2)$

 No comments.

## 2. Recursive Prediction

 Correct

Multiple Choice

### Question description

Given the following code, what is the value of `output`?

```
def mystery_function(n):  
    if n == 0:  
        return 0  
  
    if n % 2 == 1:  
        return 2 + mystery_function(n - 1)  
    else:  
        return 3 + mystery_function(n - 1)  
  
output = mystery_function(2)
```

### Candidate's Solution

**Options:** (Expected answer indicated with a tick)

☐ 0

☐ 2

☐ 3

☒ 5



 No comments.

### 3. Base Case Scenario

 Correct

Multiple Choice

#### Question description

Which of the following best describes the **base case** of the following recursive algorithm:

```
def sequence(n):  
    if n == 1 or n == 2:  
        return 1  
    else:  
        return sequence(sequence(n-1)) + sequence(n-sequence(n-1))
```

#### Candidate's Solution

Options: (Expected answer indicated with a tick)

☐ `n == 1`☐ `n == 2`☒ `n == 1 or n == 2`☐ `sequence(sequence(n-1)) + sequence(n-sequence(n-1))` No comments.

#### 4. Predict the Recursive Call

 Correct

Multiple Choice

##### Question description

Given the following code, what is the value of `output`?

```
def recursive_function(a, b):  
    if b < 0:  
        return -1 * recursive_function(a, -b)  
    if b == 0:  
        return 0  
  
    return a + recursive_function(a, b - 1)  
  
output = recursive_function(3, 2)
```

##### Candidate's Solution

Options: (Expected answer indicated with a tick)

☐ 0☐ 3☒ 6☐ -6

⚠ No comments.

## 5. String Search

✎ Correct

Coding

### Question description

Given an alphabetically sorted list of strings `names` and a string `val`, return the index of the first occurrence of `val` in  $O(\log n)$  time.

If `val` is not a name in `names`, return `-1`.

You may not use the built-in `index()` function.

Example 1:

Input: `names = ['Amal', 'Beric', 'Florin', 'Julie', 'Qin']`, `val = 'Julie'`

Output: 3

Example 2:

Input: `names = ['Amal', 'Beric', 'Florin', 'Julie', 'Qin']`, `val = 'Tabrez'`

Expected Output: -1

### Candidate's Solution

Language used: Python 3

```
1 #!/bin/python3
2
3 import math
4 import os
5 import random
6 import re
7 import sys
8 import ast
9
10
11
12 #
```



```
13 # Complete the 'find_val' function below.
14 #
15 # The function is expected to return an INTEGER.
16 # The function accepts following parameters:
17 # 1. STRING_ARRAY names
18 # 2. STRING val
19 #
20
21 def find_val(names, val):
22     # Write your code here
23     left, right = 0, len(names) - 1
24
25     while left <= right:
26         mid = (left + right) // 2
27         if names[mid] == val:
28             return mid
29         elif names[mid] < val:
30             left = mid + 1
31         else:
32             right = mid - 1
33
34     return -1
35
36 if __name__ == '__main__':
37     fptr = open(os.environ['OUTPUT_PATH'], 'w')
38
39     # Read the input list as a string and safely evaluate it to a Python list
40     names = ast.literal_eval(input().strip())
41
42     # Read the value to find, also stripping the surrounding quotes
43     val = input().strip().strip('"')
44
45     # Call the find_val function with names and val
46     result = find_val(names, val)
47
48     # Write the result to the file expected by HackerRank
49     fptr.write(str(result) + '\n')
50
51     fptr.close()
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
----------	------------	------	--------	-------	------------	-------------

Testcase 0	Easy	Hidden	Success	4	0.0403 sec	11.2 KB
Testcase 1	Easy	Hidden	Success	4	0.0377 sec	11 KB
Testcase 2	Easy	Hidden	Success	4	0.0428 sec	11.1 KB
Testcase 3	Easy	Hidden	Success	4	0.0477 sec	11 KB
Testcase 4	Easy	Hidden	Success	4	0.0442 sec	11 KB

⚠ No comments.

## 6. Power Function

✍ Correct

Coding

### Question description

Given two integers,  $x$  and  $n$ , where  $n \geq 0$ , write a function `power()` that recursively computes and returns  $x^n$ .

### Candidate's Solution

Language used: Python 3

```
1 #!/bin/python3
2
3 import math
4 import os
5 import random
6 import re
```

```
7 import sys
8
9
10
11 #
12 # Complete the 'power' function below.
13 #
14 # The function is expected to return an INTEGER.
15 # The function accepts following parameters:
16 # 1. INTEGER x
17 # 2. INTEGER n
18 #
19
20 def power(x, n):
21     # Write your code here
22     if n == 0:
23         return 1
24     if n == 1:
25         return x
26     if n % 2 == 0:
27         pow = power(x, n // 2)
28         return pow * pow
29     else:
30         pow = power(x, (n - 1) // 2)
31         return x * pow * pow
32
33
34 if __name__ == '__main__':
35     fptr = open(os.environ['OUTPUT_PATH'], 'w')
36
37     x = int(input().strip())
38
39     n = int(input().strip())
40
41     result = power(x, n)
42
43     fptr.write(str(result) + '\n')
44
45     fptr.close()
46
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
----------	------------	------	--------	-------	------------	-------------

Testcase 0	Easy	Hidden	Success	4	0.0348 sec	10.5 KB
Testcase 1	Easy	Hidden	Success	4	0.0378 sec	10.5 KB
Testcase 2	Easy	Hidden	Success	4	0.0426 sec	10.7 KB
Testcase 4	Easy	Hidden	Success	4	0.0347 sec	10.5 KB
Testcase 5	Easy	Hidden	Success	4	0.0331 sec	10.7 KB

🚫 No comments.

## 7. Find the Domain

📌 Correct

Coding

### Question description

Given a list of integers `nums` sorted in non-decreasing order, find the starting and ending position of a given `target` value. Return the result as a list in the form `[start_position, end_position]`.

If `target` is not found in the list, return `[-1, -1]`. Your solution must have  $O(\log n)$  time complexity.

Example 1:

Input: `nums = [5,7,7,8,8,10]`, `target = 8`

Output: `[3,4]`

Example 2:

Input: nums = [5,7,7,8,8,10], target = 6  
Output: [-1,-1]

## Candidate's Solution

Language used: Python 3

```
1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8  import ast
9
10
11
12
13  #
14  # Complete the 'search' function below.
15  #
16  # The function is expected to return an INTEGER_ARRAY.
17  # The function accepts following parameters:
18  # 1. INTEGER_ARRAY nums
19  # 2. INTEGER target
20  #
21
22  def search(nums, target):
23      # Write your code here
24      def find_first(nums, target):
25          left, right = 0, len(nums) - 1
26          first_pos = -1
27
28          while left <= right:
29              mid = (left + right) // 2
30              if nums[mid] == target:
31                  first_pos = mid
32                  right = mid - 1
33              elif nums[mid] < target:
34                  left = mid + 1
35              else:
36                  right = mid - 1
37          return first_pos
38
```

```
39     def find_last(nums, target):
40         left, right = 0, len(nums) - 1
41         last_pos = -1
42         while left <= right:
43             mid = (left + right) // 2
44             if nums[mid] == target:
45                 last_pos = mid
46                 left = mid + 1 # Move right to find the last occurrence
47             elif nums[mid] < target:
48                 left = mid + 1
49             else:
50                 right = mid - 1
51         return last_pos
52
53     start_pos = find_first(nums, target)
54     if start_pos == -1:
55         return [-1, -1]
56
57     end_pos = find_last(nums, target)
58     return [start_pos, end_pos]
59
60 if __name__ == '__main__':
61     fptr = open(os.environ['OUTPUT_PATH'], 'w')
62
63     nums = ast.literal_eval(input().strip())
64     target = int(input().strip())
65
66     result = search(nums, target)
67
68     # Change here: Convert the entire list to a string that looks like a list
69     fptr.write(str(result) + '\n')
70
71     fptr.close()
72
73
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Hidden	Success	4	0.0444 sec	11 KB

Testcase 1	Easy	Hidden	Success	4	0.0482 sec	11 KB
Testcase 2	Easy	Hidden	Success	4	0.0469 sec	11.3 KB
Testcase 3	Easy	Hidden	Success	4	0.0383 sec	10.9 KB
Testcase 4	Easy	Hidden	Success	4	0.0513 sec	11 KB

 No comments.