



Rajasekhar Reddy Kolagotla
Other

PDF generated at: 30 Jun 2025 01:02:21 UTC

[View this report on HackerRank](#)

Score

100% • 80 / 80
scored in TIP102: Unit 4 Version A (Standard) - Summer 2025 in 23 min 17 sec on 29 Jun 2025 17:37:08 PDT

Candidate Information

| | |
|-------------------------------|---|
| Email | rajasekhar1131997@gmail.com |
| Test | TIP102: Unit 4 Version A (Standard) - Summer 2025 |
| Candidate Packet | View |
| Taken on | 29 Jun 2025 17:37:08 PDT |
| Time taken | 23 min 17 sec/ 90 min |
| Personal Member ID | 126663 |
| Email Address with CodePath | rajasekhar1131997@gmail.com |
| Github username with CodePath | Rajasekhar1131997 |
| Invited by | CodePath |

Suspicious Activity detected

Code similarity

 Code similarity • 1 question

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

Coding Questions • 60 / 60

| Status | No. | Question | Time Taken | Skill | Score | Code Quality |
|--------|-----|--------------------|--------------------|-------|-------|--------------|
| | 1 | Move Zeroes Coding | 3 min 18 sec | - | 20/20 | - |

| | | | | | | |
|---|---|---|--------------|---|-------|---|
|  | 2 | Check if Array Is Sorted and Rotated Coding | 4 min 33 sec | - | 20/20 | - |
|  | 3 | Subarray Sum Equals K Coding | 5 min 4 sec | - | 20/20 | - |

Multiple Choice + Debugging • 20 / 20

| Status | No. | Question | Time Taken | Skill | Score | Code Quality |
|---|-----|---|--------------|-------|-------|--------------|
|  | 4 | What is the time complexity of complex_function()? Multiple Choice | 1 min 41 sec | - | 5/5 | - |
|  | 5 | Which of the following data structures/algorithmic techniques would be most appropriate for implementing this function? Multiple Choice | 1 min 21 sec | - | 5/5 | - |
|  | 6 | What is the output of the following code? Multiple Choice | 2 min 49 sec | - | 5/5 | - |
|  | 7 | Fing the bug Coding | 3 min 2 sec | - | 5/5 | - |

Coding

Question description

Given a 0-indexed integer array `nums` of length `n` and an integer `target`, write a function `count_pairs()` that returns the number of pairs (i, j) where $0 \leq i < j < n$, and $nums[i] + nums[j] < target$

Input: `nums = [-1,1,2,3,1]`, `target = 2`

Output: 3

Explanation: There are 3 pairs of indices that satisfy the conditions in the statement:

- (0, 1) since $0 < 1$ and $nums[0] + nums[1] = 0 < target$
- (0, 2) since $0 < 2$ and $nums[0] + nums[2] = 1 < target$
- (0, 4) since $0 < 4$ and $nums[0] + nums[4] = 0 < target$

Note that (0, 3) is not counted since $nums[0] + nums[3]$ is not strictly less than the target.

Candidate's Solution

Language used: Python 3

```
1  #!/bin/python
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 'count_pairs' function below.
14 #
15 # The function is expected to return an INTEGER.
16 # The function accepts following parameters:
17 # 1. INTEGER_ARRAY nums
18 # 2. INTEGER target
19 #
20
```

```

21 def count_pairs(nums, target):
22     new_nums = sorted(nums)
23     count = 0
24     left = 0
25     right = len(new_nums) - 1
26     while left < right:
27         if new_nums[left] + new_nums[right] < target:
28             count += (right-left)
29             left += 1
30         else:
31             right -= 1
32     return count
33
34 if __name__ == '__main__':
35     outfile = open(os.environ['OUTPUT_PATH'], 'w')
36     input_data = sys.stdin.read().strip().splitlines()
37
38     results = []
39
40     for line in input_data:
41         # Convert the line to list of lists
42         nums, target = eval(line)
43         result = count_pairs(nums, target)
44         results.append(result)
45
46     for res in results:
47         outfile.write(str(res) + '\n')
48     outfile.close()

```

| TESTCASE | DIFFICULTY | TYPE | STATUS | SCORE | TIME TAKEN | MEMORY USED |
|-------------------------------|------------|--------|---------|-------|------------|-------------|
| Testcase 0 | Easy | Hidden | Success | 0 | 0.0294 sec | 10.9 KB |
| Single Pair Less Than Target | Easy | Hidden | Success | 0 | 0.0297 sec | 10.8 KB |
| All Elements Form Valid Pairs | Easy | Hidden | Success | 0 | 0.0283 sec | 10.9 KB |

| | | | | | | |
|---|------|--------|---------|----|------------|---------|
| No Valid Pairs | Easy | Hidden | Success | 0 | 0.0271 sec | 10.9 KB |
| Empty Array | Easy | Hidden | Success | 0 | 0.0282 sec | 10.6 KB |
| Single Element | Easy | Hidden | Success | 0 | 0.0286 sec | 10.9 KB |
| All Elements Greater Than or Equal to Target | Easy | Hidden | Success | 0 | 0.028 sec | 10.8 KB |
| All Elements Less Than Target, Multiple Occurrences | Easy | Hidden | Success | 0 | 0.0266 sec | 10.9 KB |
| Duplicates with Same Result | Easy | Hidden | Success | 0 | 0.0288 sec | 10.9 KB |
| Pass/Fail Case | Easy | Hidden | Success | 20 | 0.0301 sec | 10.8 KB |

No comments.

2. Check if Array Is Sorted and Rotated

Correct

Coding

Question description

Given an array `nums`, return `True` if the array was originally sorted in non-decreasing order, then rotated **some** number of positions (including zero). Otherwise, return `False`.

There may be **duplicates** in the original array.

Note: Rotating an array by a certain number of positions means shifting each element that many positions to the right (or left). When shifted one position to the right, the last element in the array becomes the first element in the array. Formally, an array `A` rotated by `x` positions results in an array `B` of the same length such that `A[i] == B[(i+x) % A.length]`, where `%` is the modulo operation.

Example 1:

Input: `nums = [3,4,5,1,2]`

Output: `True`

Explanation: `[1,2,3,4,5]` is the original sorted array.

You can rotate the array by `x = 3` positions to begin on the the element of value 3: `[3,4,5,1,2]`.

Example 2:

Input: `nums = [2,1,3,4]`

Output: `False`

Explanation: There is no sorted array once rotated that can make `nums`.

Example 3:

Input: `nums = [1,2,3]`

Output: `true`

Explanation: `[1,2,3]` is the original sorted array.

You can rotate the array by `x = 0` positions (i.e. no rotation) to make `nums`.

Candidate's Solution

Language used: Python 3

```
1 #!/bin/python
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 'is_sorted_rotated' function below.
14 #
15 # The function is expected to return a BOOLEAN.
16 # The function accepts INTEGER_ARRAY nums as parameter.
17 #
18
19 def is_sorted_rotated(nums):
20     if len(nums) < 2:
21         return True
22     drops = 0
23     for i in range(len(nums)):
24         if nums[i] > nums[(i+1) % len(nums)]:
25             drops += 1
26             if drops > 1:
27                 return False
28     return True
29
30
31 if __name__ == '__main__':
32     outfile = open(os.environ['OUTPUT_PATH'], 'w')
33     input_data = sys.stdin.read().strip().split('\n')
34
35     for line in input_data:
36         nums = ast.literal_eval(line.strip())
37         result = is_sorted_rotated(nums)
38         outfile.write(str(result) + '\n')
39     outfile.close()

```

| TESTCASE | DIFFICULTY | TYPE | STATUS | SCORE | TIME TAKEN | MEMORY USED |
|--------------------------------|------------|--------|---------|-------|------------|-------------|
| Basic Case | Easy | Hidden | Success | 0 | 0.0381 sec | 10.9 KB |
| Standard Case without Rotation | Easy | Hidden | Success | 0 | 0.0273 sec | 10.9 KB |
| No Rotation Needed | Easy | Hidden | Success | 0 | 0.0342 sec | 10.6 KB |

| | | | | | | |
|------------------------------------|------|--------|---------|----|------------|---------|
| Single Element | Easy | Hidden | Success | 0 | 0.0296 sec | 10.9 KB |
| Two Elements Sorted | Easy | Hidden | Success | 0 | 0.0274 sec | 10.9 KB |
| Two Elements Not Sorted | Easy | Hidden | Success | 0 | 0.0313 sec | 10.9 KB |
| All Identical Elements | Easy | Hidden | Success | 0 | 0.029 sec | 10.9 KB |
| Array Already Sorted with Rotation | Easy | Hidden | Success | 0 | 0.028 sec | 10.9 KB |
| Empty Array | Easy | Hidden | Success | 0 | 0.0285 sec | 11 KB |
| Pass/Fail Case | Easy | Hidden | Success | 20 | 0.0374 sec | 10.8 KB |

No comments.

3. Subarray Sum Equals K

Correct

Coding

Question description

Given an array of integers `nums` and an integer `k`, return the total number of continuous subarrays whose sum equals to `k`.

Example 1:

Input: [1,1,1], 2

Output: 2

Example 2:

Input: [1, 2, 3], 7

Output: 0

Candidate's Solution

Language used: Python 3

```
1  #!/bin/python
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8  import ast
9  import json
10
11
12
13  #
14  # Complete the 'subarray_sum' function below.
15  #
16  # The function is expected to return an INTEGER.
17  # The function accepts following parameters:
18  # 1. INTEGER_ARRAY nums
19  # 2. INTEGER k
20  #
21
22  def subarray_sum(nums, k):
23      frequency = {0:1}
24      running_sum = 0
25      count = 0
26      for num in nums:
27          running_sum += num
28          count += frequency.get(running_sum-k, 0)
29          frequency[running_sum] = frequency.get(running_sum, 0) + 1
30      return count
```

```

31
32 if __name__ == '__main__':
33     outfile = open(os.environ['OUTPUT_PATH'], 'w')
34     input_data = sys.stdin.read().strip().splitlines()
35
36     results = []
37
38     for line in input_data:
39         parts = json.loads(f"[{line}]")
40         nums = parts[0]
41         k = parts[1]
42         result = subarray_sum(nums, k)
43         results.append(result)
44
45     for res in results:
46         outfile.write(str(res) + '\n')
47     outfile.close()

```

| TESTCASE | DIFFICULTY | TYPE | STATUS | SCORE | TIME TAKEN | MEMORY USED |
|---------------------------------|------------|--------|---------|-------|------------|-------------|
| Basic Case | Easy | Hidden | Success | 0 | 0.0293 sec | 10.9 KB |
| Target Sum Found | Easy | Hidden | Success | 0 | 0.0305 sec | 11 KB |
| Multiple Subarrays for Target 0 | Easy | Hidden | Success | 0 | 0.0307 sec | 10.9 KB |
| Single Element Equal to k | Easy | Hidden | Success | 0 | 0.032 sec | 11 KB |
| No Valid Subarrays | Easy | Hidden | Success | 0 | 0.0324 sec | 11 KB |

| | | | | | | |
|---|------|--------|---------|----|------------|---------|
| Negative Numbers | Easy | Hidden | Success | 0 | 0.0543 sec | 11 KB |
| All Zeros | Easy | Hidden | Success | 0 | 0.0307 sec | 11 KB |
| Target Sum Greater than Sum of All Elements | Easy | Hidden | Success | 0 | 0.0323 sec | 10.8 KB |
| Large k Value | Easy | Hidden | Success | 0 | 0.0324 sec | 11 KB |
| Alternating Positive and Negative Values | Easy | Hidden | Success | 0 | 0.0293 sec | 10.9 KB |
| All Negative Elements, Target is Negative | Easy | Hidden | Success | 0 | 0.0315 sec | 10.9 KB |
| Pass/Fail Case | Easy | Hidden | Success | 20 | 0.03 sec | 11 KB |

🚫 No comments.

4. What is the time complexity of `complex_function()`?

✅ Correct

Multiple Choice

Question description

What is the time complexity of `complex_function()`?

```
def complex_function(nums):  
    total = 0  
    for i in range(len(nums)):  
        for j in range(i, len(nums)):  
            if nums[i] < nums[j]:  
                total += nums[j] - nums[i]  
    return total
```

Candidate's Solution

Options: (Expected answer indicated with a tick)

☐ $O(1)$

☐ $O(\log n)$

☐ $O(n)$

☒ $O(n^2)$



 No comments.

5. Which of the following data structures/algorithmic techniques would be most appropriate for implementing this function?

 Correct

Multiple Choice

Question description

The function `find_mode()` takes in a list of elements `lst` and returns the mode of the the list. If there are multiple modes, it returns them all as a list. If there is a single mode it returns the mode as a single value. The mode is the element that occurs most often in the list.

Which of the following data structures/algorithmic techniques would be most appropriate for implementing this function?

```
# Example usage:
nums1 = [1, 2, 2, 3, 4]
print(find_mode(nums1)) # Output: 2

nums2 = [1, 2, 2, 3, 3, 4]
print(find_mode(nums2)) # Output: [2, 3]

nums3 = [1, 2, 3, 4, 5]
print(find_mode(nums3)) # Output: [1, 2, 3, 4, 5] (all values are unique)
```

Candidate's Solution

Options: (Expected answer indicated with a tick)

☐ stack☐ queue☐ two pointer



frequency map



No comments.

6. What is the output of the following code?

 Correct

Multiple Choice

Question description

```
def process_numbers(nums, threshold):  
    stack = []  
    for num in nums:  
        if num < threshold:  
            stack.append(num)  
        elif num <= 10 and stack:  
            stack.pop()  
    return stack  
  
print(process_numbers([3, 5, 1, 9, 6, 15], 8))
```

Candidate's Solution

Options: (Expected answer indicated with a tick)



[5, 1, 6]



[3, 5, 6]



☐ [3, 5, 1, 6]☐ [3, 5, 1]

⚠ No comments.

7. Fing the bug

✓ Correct

Coding

Question description

The following code incorrectly implements `is_anagram()`. When implemented correctly, `is_anagram()` should accept two strings `s` and `t` and return `True` if `s` and `t` are anagrams of each other. Otherwise, it should return `False`.

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once.

Identify any bug(s) within the given implementation and correct the code so that it successfully passes the provided test cases.

Candidate's Solution

Language used: Python 3

```
1 #!/bin/python
2
3 import math
4 import os
5 import random
6 import re
7 import sys
8 import ast
9 def is_anagram(s, t):
10     if len(s) != len(t):
11         return False
```



```
12
13     s_freq = {}
14     t_freq = {}
15
16     for char in s:
17         s_freq[char] = s_freq.get(char, 0) + 1
18
19     for char in t:
20         t_freq[char] = t_freq.get(char, 0) + 1
21
22     return s_freq == t_freq
23 if __name__ == '__main__':
24     input_data = sys.stdin.read().strip()
25     input_list = ast.literal_eval(input_data)
26
27     s = input_list[0]
28     t = input_list[1]
29
30     result = is_anagram(s, t)
31     print(result)
```

| TESTCASE | DIFFICULTY | TYPE | STATUS | SCORE | TIME TAKEN | MEMORY USED |
|----------------|------------|--------|---------|-------|------------|-------------|
| Pass/Fail Case | Easy | Hidden | Success | 5 | 0.026 sec | 10.8 KB |

No comments.