

Mohammed Basith

Candidate

E-mail:
basithpersonel@icloud.com

Session

ID: CFU3C5-XAZ
Time limit: 70 min.
Report recipients: No one
Accessed from: 136.185.96.205,
136.185.96.205
Invited by: arun.k@uplers.in

Status: completed

Created on: 2022-11-27 10:25 UTC
Started on: 2022-11-27 10:25 UTC
Finished on: 2022-11-27 10:32 UTC

Notes:

N/A

Similarity Check

Status: not found
No similar solutions have been detected.

Test score

100%

Tasks in test

1 | AppleOrchardCoding
Submitted in: Python

Score

100%

Tasks Details

Medium

1. AppleOrchardCoding

Compute the maximum number of apples that can be picked by two people during their work in the orchard.

Task Score	Correctness	Performance
100	100	Not assessed

Task description

Alice and Bob work in a beautiful orchard. There are N apple trees in the orchard. The apple trees are arranged in a row and they are numbered from 1 to N .

Alice is planning to collect all the apples from K consecutive trees and Bob is planning to collect all the apples from L consecutive trees. They want to choose two disjoint segments (one consisting of K trees for Alice and the other consisting of L trees for Bob) so as not to disturb each other. What is the maximum number of apples that they can collect?

Write a function:

```
def solution(A, K, L)
```

that, given an array A consisting of N integers denoting the number of apples on each apple tree in the row, and integers K and L denoting, respectively, the number of trees that Alice and Bob can choose when collecting, returns the maximum number of apples that can be collected by them, or -1 if there are no such intervals.

For example, given $A = [6, 1, 4, 6, 3, 2, 7, 4]$, $K = 3$, $L = 2$, your function should return 24, because Alice can choose trees 3 to 5 and collect $4 + 6 + 3 = 13$ apples, and Bob can choose trees 7 to 8 and collect $7 + 4 = 11$ apples. Thus, they will collect $13 + 11 = 24$ apples in total, and that is the maximum number that can be achieved.

Given $A = [10, 19, 15]$, $K = 2$, $L = 2$, your function should return -1 , because it is not possible for Alice and Bob to choose two disjoint intervals.

Assume that:

- N is an integer within the range $[2..100]$;
- K and L are integers within the range $[1..N - 1]$;
- each element of array A is an integer within the range $[1..500]$.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution
[See Live Version](#)

Programming language used: Python

Total time used: 7 minutes

?

Effective time used: 7 minutes

?

Notes: *not defined yet*

Source code

Code: 10:32:27 UTC, py, final, score: 100

```
1# you can write to stdout for debugging purposes, e.g.
2# print("this is a debug message")
3
4def solution(A, K, L):
5    # write your code in Python 3.8.10
6    first = findMaxApples(A, K, L)
7    second = findMaxApples(A, L, K)
8    return max(first, second)
9
10def findMaxApples(A, K, L):
11    if K + 1 > len(A):
```

```

12     return -1
13     sum = [0] * (1000)
14     sum[0] = A[0]
15     # for i in range(len(A),1):
16     #     sum.append(sum[i-1]+A[i])
17     i = 1
18     while (i < len(A)):
19         sum[i] = sum[i - 1] + A[i]
20         i += 1
21     max = -1
22     x, y, a = 0, 0, 0
23     while (a + K - 1 < len(A)):
24         if (a > 0):
25             x = sum[a + K - 1] - sum[a - 1]
26         else:
27             x = sum[a + K - 1]
28         b = a + K
29         while (b + L - 1 < len(A)):
30             if (b > 0):
31                 y = sum[b + L - 1] - sum[b - 1]
32             else:
33                 y = sum[b + L - 1]
34             if (x + y > max):
35                 max = x + y
36             b += 1
37         a += 1
38     return max

```

Analysis summary

The solution obtained perfect score.

Analysis

Example tests	
example1 First example test.	✓ OK
example2 Second example test.	✓ OK
Correctness tests	
simple_tests Simple tests. N <= 10.	✓ OK
simple_almost_all_apples Tests where all but few apples are available. N <= 10.	✓ OK
random_test_possibles_almost_all_apples Random tests where all but few apples are available. N <= 50.	✓ OK
random_possible Random tests where the answer is not -1. N <= 50.	✓ OK
random_test_possibles_all_apples Random tests where all apples are available.	✓ OK
random Random tests.	✓ OK
random_not_possible Random tests where the answer is -1.	✓ OK