

CodeCheck Report: training5PDN3H-JT9

[Check out Codility training tasks](#)

Test Name:

SummaryTimeline

Tasks summary

Task	Time spent	Score
<div><div>TieRopes</div><div>Java 8</div><div></div></div>	8 min	100%

Total score

100%

Tasks Details

Easy	1. TieRopes	Task Score	Correctness	Performance
	Tie adjacent ropes to achieve the maximum number of ropes of length $\geq K$.			
		100%	100%	100%

Task description

There are N ropes numbered from 0 to $N - 1$, whose lengths are given in an array A , lying on the floor in a line. For each I ($0 \leq I < N$), the length of rope I on the line is $A[I]$.

We say that two ropes I and $I + 1$ are *adjacent*. Two adjacent ropes can be tied together with a knot, and the length of the tied rope is the sum of lengths of both ropes. The resulting new rope can then be tied again.

For a given integer K , the goal is to tie the ropes in such a way that the number of ropes whose length is greater than or equal to K is maximal.

For example, consider $K = 4$ and array A such that:

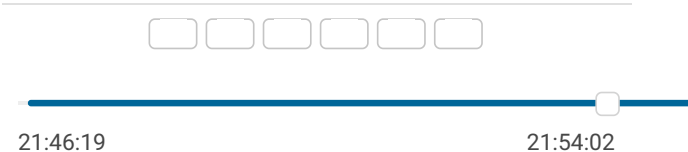
```
A[0] = 1
A[1] = 2
A[2] = 3
A[3] = 4
A[4] = 1
A[5] = 1
A[6] = 3
```

The ropes are shown in the figure below.

Solution

Programming language used:	Java 8
Total time used:	8 minutes?
Effective time used:	8 minutes?
Notes:	not defined yet

Task timeline?



Code: 21:54:01 UTC, java, [show code in pop-up](#)
final, score: 100

1

// you can also use imports, for example:

2

// import java.util.*;



We can tie:

- rope 1 with rope 2 to produce a rope of length $A[1] + A[2] = 5$;
- rope 4 with rope 5 with rope 6 to produce a rope of length $A[4] + A[5] + A[6] = 5$.

After that, there will be three ropes whose lengths are greater than or equal to $K = 4$. It is not possible to produce four such ropes.

Write a function:

```
class Solution { public int solution(int K,
int[] A); }
```

that, given an integer K and a non-empty array A of N integers, returns the maximum number of ropes of length greater than or equal to K that can be created.

For example, given $K = 4$ and array A such that:

- $A[0] = 1$
- $A[1] = 2$
- $A[2] = 3$
- $A[3] = 4$
- $A[4] = 1$
- $A[5] = 1$
- $A[6] = 3$

the function should return 3, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range $[1..100,000]$;
- K is an integer within the range $[1..1,000,000,000]$;
- each element of array A is an integer within the range $[1..1,000,000,000]$.

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
3
4 // you can write to stdout for debugging purposes
5 // System.out.println("this is a debug message");
6
7 class Solution {
8     public int solution(int K, int[] A) {
9         int N = A.length;
10
11         int count = 0;
12
13         int currentLength = 0;
14         for (int i = 0; i < N; i++) {
15             currentLength += A[i];
16             if (currentLength >= K) {
17                 count++;
18                 currentLength = 0;
19             }
20         }
21         return count;
22     }
23 }
24 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **$O(N)$**

collapse all		Example tests
▼	example	✓ OK
	example test	
1. 0.004 s		OK
collapse all		Correctness tests
▼	single	✓ OK
	single element	
1. 0.004 s		OK
2. 0.004 s		OK
3. 0.004 s		OK
4. 0.004 s		OK
5. 0.004 s		OK
▼	double	✓ OK
	two elements	
1. 0.008 s		OK
2. 0.004 s		OK
3. 0.004 s		OK
4. 0.004 s		OK
5. 0.004 s		OK
▼	small_functional	✓ OK
	small functional tests	
1. 0.008 s		OK

2.	0.004 s	OK
▼	small_random	✓ OK
	small random sequences length = ~100	
1.	0.004 s	OK
2.	0.008 s	OK
collapse all Performance tests		
▼	medium_random	✓ OK
	chaotic medium sequences length = ~5,000	
1.	0.024 s	OK
2.	0.012 s	OK
3.	0.012 s	OK
▼	large_range	✓ OK
	large range test, length = ~100,000	
1.	0.168 s	OK
2.	0.168 s	OK
3.	0.280 s	OK
▼	large_answer	✓ OK
	test with large answer, length = ~100,000	
1.	0.432 s	OK
2.	0.424 s	OK
3.	0.244 s	OK
4.	0.400 s	OK
▼	small_answer	✓ OK
	test with large answer, length = ~100,000	
1.	0.164 s	OK
2.	0.168 s	OK
3.	0.076 s	OK
4.	0.168 s	OK