Codility_

CodeCheck Report: trainingHDVH4V-K7D

Test Name:

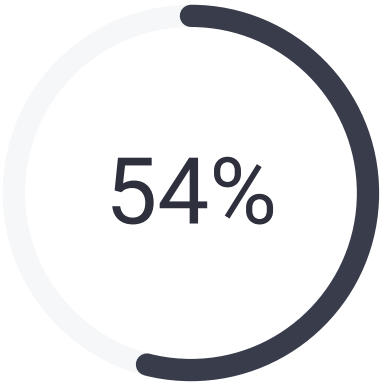
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
MinAbsSum Python	11 min	54%

Total score



Tasks Details

1. **MinAbsSum**

Hard

Given array of integers, find the lowest absolute sum of elements.

Task Score	Correctness	Performance
54%	100%	0%

Task description

For a given array A of N integers and a sequence S of N integers from the set {−1, 1}, we define val(A, S) as follows:

val(A, S) = |**sum**{ A[i]*S[i] for i = 0..N−1 }|

(Assume that the sum of zero elements equals zero.)

For a given array A, we are looking for such a sequence S that minimizes val(A,S).

Write a function:

```
def solution(A)
```

that, given an array A of N integers, computes the

Solution

Programming language used:	Python
Total time used:	11 minutes
Effective time used:	11 minutes
Notes:	*not defined yet*

Task timeline

minimum value of val(A,S) from all possible values of val(A,S) for all possible sequences S of N integers from the set {−1, 1}.

For example, given array:

```
A[0] =  1
A[1] =  5
A[2] =  2
A[3] = −2
```

your function should return 0, since for S = [−1, 1, −1, 1], val(A, S) = 0, which is the minimum possible value.

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

- N is an integer within the range [0..20,000];
- each element of array A is an integer within the range [−100..100].

12:58:30 13:08:44

Code: 13:08:43 UTC, py, final, score: **54** show code in pop-up

```
1  # you can write to stdout for debugging
2  # print("this is a debug message")
3
4  def solution(A):
5      # Implement your solution here
6      # pass
7      N = len(A)
8      min_val = float('inf')
9
10     # Iterate over all possible binary
11     for i in range(2**N):
12         sequence = [1 if (i >> j) & 1 e
13         value = abs(sum(A[j] * sequence
14         min_val = min(min_val, value)
15
16     return min_val
17
18
19
20
21
22
23
24
25
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis

Detected time complexity: **O(N**2 * max(abs(A)))**

expand all	Example tests	
▶	example1 example test	✔ OK
expand all	**Correctness tests**	
▶	simple1 simple 1	✔ OK
▶	simple2 simple 2	✔ OK
▶	simple3 simple 3	✔ OK
▶		

range range 2..20	✔ OK
▶ extreme empty and single element	✔ OK
▶ functional small functional test	✔ OK

expand all **Performance tests**

▶ medium1 medium random	✘ TIMEOUT ERROR Killed. Hard limit reached: 6.000 sec.
▶ medium2 multiples of 10 + 5	✘ TIMEOUT ERROR Killed. Hard limit reached: 6.000 sec.
▶ big1 multiples of 5 + 42	✘ TIMEOUT ERROR Killed. Hard limit reached: 9.000 sec.
▶ big3 all 4s and one 3	✘ TIMEOUT ERROR Killed. Hard limit reached: 6.000 sec.
▶ big4 multiples of 10	✘ TIMEOUT ERROR Killed. Hard limit reached: 11.000 sec.