Find k elements summing to zero

Serge Kruk

October 9, 2024

1 Problem statement

Given an (unsorted) array of n integers and a positive integer k, return k elements of the array that sum to zero, if there are such k elements. Otherwise return `None`.

For those of you who know Python's library very well: Yes, I know there is a library function that will do most of the work for you, but the whole point of this class is not to uncover work done by others, but rather to develop the algorithms yourself. So, you cannot call any library function other than sorting, if you need to. Since we have developped sorting algorithms already, that is allowed.

This is not a difficult assignment. You should be able to come up with a solution almost immediately. But pay particular attention to your runtime computations and proof of correctness.

To clarify any misunderstanding on the correct parameters, here is the function"s signature. Fill-in the blanks.

```python
def k_zero_sum(a,k):
   """a is an array of n elements.
      We return an array of the k elements summing to 0."""
   return None
```

1.1 Testing help

To help you with testing, here is a test function. Once you have developped your code, run this to see if it passes this minimal test. If not, you certainly have errors in your code. If it does, you should create more exhaustive test. Include this test and its result in your report, of course.

```python
from random import randint, shuffle
def valid(a,k,b):
  # We we check that the array b sums to zero, is of the proper length and is from a.
  return all([len(b)==k]+[sum(b)==0]+[e in a for e in b])
def test_k_zero_sum():
  # First we will create an array with subarrays of various sizes summing to zero.
  a,start,end = [],2,5
  for k in range(start-1,end):
    b = [randint(-100,100) for _ in range(k)] # k-1 random integers
    b.append(-sum(b))                         # one more, to make the sum 0
    a = a+b                                   # append to our test array
  shuffle(a)                                  # mix it up
  # Now that we have this array, we call k_zero_sum for each k.
  # If correct, it must find each of the subarrays.
  return all([valid(a,k,k_zero_sum(a,k)) for k in range(start,end+1)])
```

```python
print(test_k_zero_sum())
```