Solution Review: Find Two Numbers that Add up to "k"

This review provides a detailed analysis of the different ways to solve the Find Two Numbers that Add Up To k Challenge.

We'll cover the following

Solution #1: Using a Dictionary

```
    foundValues = {}
    for ele in lst:
        # Check for value in dictionary
        # If found return
        try:
            foundValues[k - ele]
            return [k - ele, ele]
        except KeyError:
            foundValues[ele] = 0
    return "No numbers add upto k"


print(findSum([1, 3, 2, 4], 6))

```

The best way to solve this problem is to insert every element into a dictionary. This takes O(1) as constant time insertion.

Then, for every element x in the list, we can just look up its complement, $k-x$, and, if found, return both $k-x$ and x.

Time Complexity

Each lookup is a constant time operation. Overall the running time of this approach is $O(n)$.

Solution #2: Using the Python `set()`

```
1  def findSum(lst, value):
2      foundValues = set()
3      for ele in lst:
4          if value - ele in foundValues:
5              return [value-ele, ele]
6          foundValues.add(ele)
7      return False
8
9
10 print(findSum([1, 2, 3, 4], 6))
11
```

This solution does the same thing as solution #1 except that it uses Python's built-in `set()` which makes `foundValues` an iterable sequence like a dictionary. Note that `set.add` method adds an element if element is not present in the set as in **line 6**.

Time Complexity

The time complexity of the solution above is $O(n)$.

Back

Challenge 8: Find Two Numbers that ...

Next

Challenge 9: First Non-Repeating Inte...

Mark as Completed

Report an Issue