Solution Review: A List as a Subset of Another List

This review provides a detailed analysis of the solution to the A List as a Subset of Another List Challenge.

We'll cover the following

Solution: Lookup in a Hash Table

```
def is_subset(list1, list2):
    s = set(list1)  # Create a set with list1 values
    # Traverse list 2 elements
    for elem in list2:
        # Return false if an element not in list1
        if elem not in s:
            return False
    # Return True if all elements in list1
    return True


list1 = [9, 4, 7, 1, -2, 6, 5]
list2 = [7, 1, -2]
list3 = [10, 12]
print(is_subset(list1, list2))
print(is_subset(list1, list3))

```

Output 0.189s

```
True
False
```

The solution is very simple when working with the Pythonic hash table `Set`. We simply iterate over `list2` and `list3` to see whether their elements can be found in `list1`.

At the back end, the values are checked against their hashed indices in `list1`.

Time Complexity

For a lookup list with **m** elements and a subset list with **n** elements, the time complexity is *O(m+n)*.

Back

Challenge 1: A List as a Subset of Ano...

Next

Challenge 2: Check if Lists are Disjoint

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/solution-review-a-list-as-a-subset-of-another-list__introduction-to-hashing__data-structures-for-coding-interviews-in-python)