Solution Review: Find Minimum Value in List

This review provides a detailed analysis of the different ways to find a minimum value in a list.

We'll cover the following

Solution #1: Sort the list

```
def find_minimum(lst):
    if (len(lst) <= 0):
        return None
    lst.sort()  # sort list
    return lst[0]  # return first element


print(find_minimum([9, 2, 3, 6]))
```

This solution sorts the list in ascending order using `.sort` function and returns the first element which is also the minimum. Also, if the list is emp

`None` is returned.

> We used the generic Python `.sort()` function here, but in a real interview, you should implement your own sort function if you're going to use this solution. Learn about the famous sorting method, Merge sort.

Let's implement the sorting function below and call that function in the `find_minimum` function:

```
def merge_sort(my_list):
    if len(my_list) > 1:
        mid = len(my_list) // 2
        left = my_list[:mid]
        right = my_list[mid:]

        # Recursive call on each half
        merge_sort(left)
        merge_sort(right)

        # Two iterators for traversing the two halves
        i = 0
        j = 0

        # Iterator for the main list
        k = 0

        while i < len(left) and j < len(right):
            if left[i] < right[j]:
              # The value from the left half has been used
              my_list[k] = left[i]
              # Move the iterator forward
              i += 1
            else:
                my_list[k] = right[j]
                j += 1
            # Move to the next slot
            k += 1
```

Time Complexity#

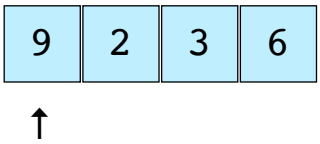
The build-in sort function `sort` and the `mergeSort` are in $O(nlogn)$. Since we only index and return after that, which are constant time operations, this solution takes $O(nlogn)$ time.

Solution #2: Iterate over the list#

```
def find_minimum(lst):
    if (len(lst) <= 0):
        return None
    minimum = lst[0]
    for ele in lst:
        # update if found a smaller element
        if ele < minimum:
            minimum = ele
    return minimum


print(find_minimum([9, 2, 3, 6]))
```

Start with the first element which is **9** in this example and save it as the smallest value. Then, iterate over the rest of the list and whenever an element that is smaller than the number already stored as `minimum` is come across, set `minimum` to that number. By the end of the list, the number stored in `minimum` will be the smallest integer in the whole list.



minimum = 9

1 of 4

Also, if the list is empty, `None` is returned.

Time Complexity#

Since the entire list is iterated over once, this algorithm is in linear time, $O(n)$.

Back

Challenge 5: Find Minimum Value in List

Next

Challenge 6: First Non-Repeating Inte...

Complete