```
def find_max(nums: list[int]) -> int:
    '''
    Finds the maximum value in the given list of integers,
    or None if the list is empty
    '''
    largest: int = None
    if len(nums) == 1:
        largest = nums[0]
    elif len(nums) >1:
        if nums[0] > nums[1]:
            largest = find_max([nums[0]] + nums[2:])
        else:
            largest = find_max(nums[1:])

    return largest



assert find_max([23, 23423, 23, 5743, 4444444]) == 4444444
assert find_max([]) == None
assert find_max([12123123, 23, 434]) == 12123123
```

Question: I believe that recursive was not the best choice in this case because although it worked, it forced the creation and use of many different versions of the function when going through the list rather than just one creation that goes through each item of the list in a loop. Not only that, the code itself was also longer and more complicated than a for loop would have been.