

Problem 1:

My solution for the `find_max` function is to compare the first two values of the list and remove the one that is lower, if the list has more than 2 items, call the function again, if not return the largest value. There is an image below showing how it works. I also added an exception in case a value other than an int is received. To answer the second part of this problem, I think using a loop is more simple to understand and faster than recursion. Also, with recursion we are dealing with a limit for the array length.

```
def find_max(nums: list[int]) -> int:

    if nums == [] or nums == None or type(nums) != list:
        return None;

    _largest = nums[0]

    print(_largest)

    try:
        if len(nums) > 1:
            if nums[1] > _largest:
                _largest = nums[1]
                nums.pop(0)
            else:
                nums.pop(1)

            if len(nums) >= 2:
                _largest = find_max(nums)

    except TypeError:
        print("The list contains a value other than ints")

    return _largest

if __name__ == "__main__":
    assert find_max([0,-1,-500,234,5,2,6,3,-500, 234]) == 234
    assert find_max([0]) == 0
    assert find_max([]) == None
    assert find_max(9) == None
    assert find_max([1000000000*-1,-1,-500,234,5,2,6,3,-500,"500"]) == 234
    assert find_max([-1,-500,-234,-5]) == -1
    assert find_max([-1,5,3]) == 5
```