1.6 MAXIMUM ELEMENT IN SORTED ARRAY

Question:

You have an algorithm that process a list of numbers. It firsts sorts the list using an efficient sorting algorithm and then finds the maximum element in sorted list. Write the code for the same.

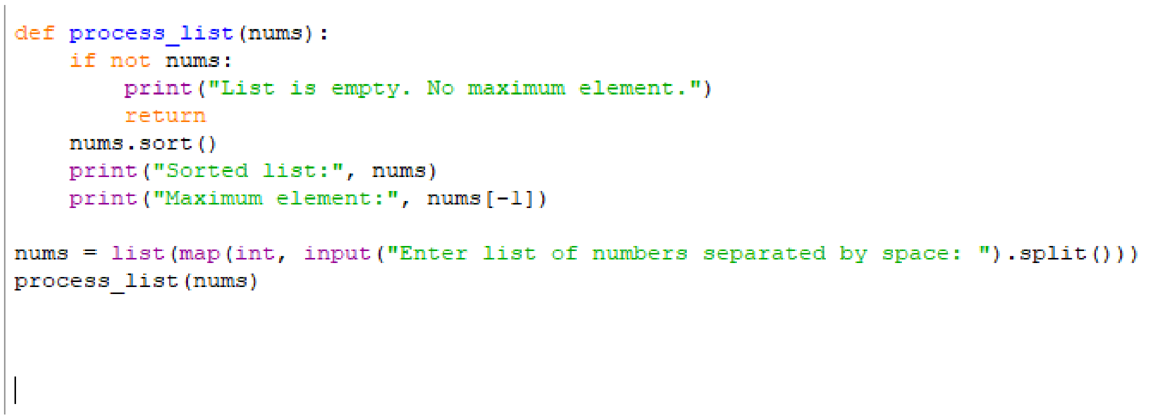
AIM:

To sort a list of integers efficiently and return the maximum element from the sorted list, while handling edge cases properly.

ALGORITHM:

1. If the list is empty, return None or print a message.
2. Sort the list using an efficient algorithm (O(n log n)).
3. Return the last element (which is the maximum).

PROGRAM:



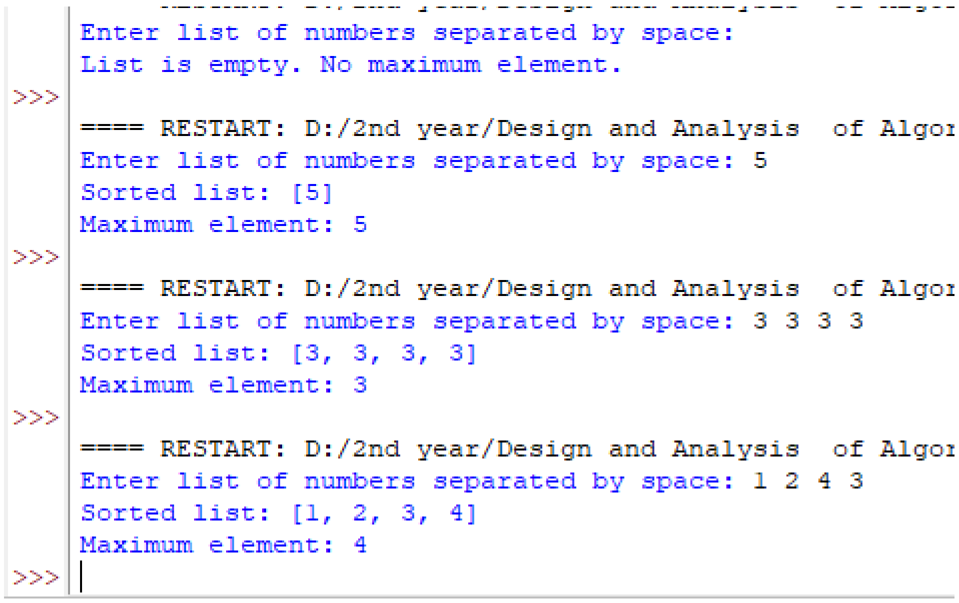
```python
def process_list(nums):
    if not nums:
        print("List is empty. No maximum element.")
        return
    nums.sort()
    print("Sorted list:", nums)
    print("Maximum element:", nums[-1])

nums = list(map(int, input("Enter list of numbers separated by space: ").split()))
process_list(nums)
```

Input:

nums = 1 2 3 4

Output:



```
Enter list of numbers separated by space:
List is empty. No maximum element.
>>>
==== RESTART: D:/2nd year/Design and Analysis  of Algor
Enter list of numbers separated by space: 5
Sorted list: [5]
Maximum element: 5
>>>
==== RESTART: D:/2nd year/Design and Analysis  of Algor
Enter list of numbers separated by space: 3 3 3 3
Sorted list: [3, 3, 3, 3]
Maximum element: 3
>>>
==== RESTART: D:/2nd year/Design and Analysis  of Algor
Enter list of numbers separated by space: 1 2 4 3
Sorted list: [1, 2, 3, 4]
Maximum element: 4
>>>
```

RESULT:

Thus the program is successfully executed, and the output is verified.

PERFORMANCE ANALYSIS:

- Sorting → O(n log n)
- Accessing last element → O(1)
- Total: O(n log n)
- Space: O(1) (in-place sort) or O(n) (depending on sorting algorithm).