1.7 CREATE A NEW LIST FROM THE ORIGINAL LIST

Question:

Write a program that takes an input list of n numbers and creates a new list containing only the unique elements from the original list. What is the space complexity of the algorithm?

AIM:

To create a program that takes an input list of n numbers and creates a new list containing only the unique elements from the original list.

ALGORITHM:

1. Initialize an empty set seen and an empty list unique_list.
2. Iterate through each element x in the input list.
3. If x is not in seen, add it to both seen and unique_list.
4. Return unique_list.

PROGRAM:

```
def get_unique_elements():
    arr = list(map(int, input("Enter list elements separated by space: ").split()))
    seen = set()
    unique = []
    for num in arr:
        if num not in seen:
            unique.append(num)
            seen.add(num)
    print("Unique elements:", unique)
get_unique_elements()
```

Input:

nums = [3, 7, 3, 5, 2, 5, 9, 2]

Output:

```
Enter list elements separated by space: 3 7 3 5 2 5 9 2
Unique elements: [3, 7, 5, 2, 9]
>>>
```

RESULT:

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

PERFORMANCE ANALYSIS:

We use a set (seen) to store unique elements → O(n) in worst case.

We also use a list (unique_list) to return result → O(n).

- Total Space Complexity = O(n)
- Time Complexity = O(n) (since each membership check in a set is O(1) on average).