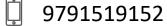


#### B.Bhuvaneswaran, AP (SG) / CSE



⊠ bhuvaneswaran@rajalakshmi.edu.in



## RAJALAKSHMI ENGINEERING COLLEGE

An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

#### Bucketing

A more efficient approach is to use the concept of bucketing.

#### Find the Duplicate Element in an Array

■ Size: n + 1

Numbers: 1 to n

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

Bucketing

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

#### **Brute Force**

- Time Complexity: O(n²)
- Space Complexity: O(1)

#### Bucketing

- For each element in the array A, we increment it's frequency (or number of occurrences) in B.
- The element which has more than one occurrence is our duplicate element.
- The time complexity of this approach is O(n).

Bucketing

However, the space complexity of this approach is also O(n), since we will always be required to have a secondary array B which stores our buckets.

#### **Bucketing Approach**

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0

#### **Bucketing Approach**

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	0	0	0	1	0	0	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	0	0	1	1	0	0	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	0	1	1	1	0	0	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	0	1	1	1	0	1	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
0	1	1	1	1	0	1	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	0	1	0	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	0	1	1	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	0	1	2	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	2	0

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	2	1

0	1	2	3	4	5	6	7	8	9
5	4	3	7	2	1	8	8	6	9

0	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	2	1

- Time Complexity: O(n)
- Space Complexity: O(n)

# Queries?

# Thank You...!