1. False

-bucket is created according to the maximum value in the array, there could be many more wasted bucket space(when the input array elements are not consecutive) which increases the running time every time the bucket is scanned .

2. {80, 27, 72, 1, 27, 8, 64, 34, 16}

Radix=9;

R[]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 27 |  |  |  |  |  |  |  |  |
| 72 | 64 |  |  |  |  |  | 16 | 8 |
| 27 | 1 |  |  |  |  |  | 34 | 80 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Q[]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 34 |  |  |  |  |  |
| 8 |  |  | 27 |  |  |  |  | 80 |
| 1 | 16 |  | 27 |  |  |  | 64 | 72 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Sorted order: {1, 8, 16, 27, 27, 34, 64, 72, 80}

3.

Algorithm FindUnique(list, range)

Input - list of integers and range of the integers

Output – find the unique element

n 🡨list of elements

Bucket 🡨 list of 3n-1 spaces, to store frequency of elements, each set to 0

For i 🡨 0 to n

Bucket[list[i]]+1

For j 🡨 0 to 3n-1

If Bucket[j] equal to 1 then return j

It is O(n) because its bucket search and range is not n^2 (range is 3n-1)