An Introduction to Bucket Sort

Amlan Saha
Dhiman Goswami

Bangladesh University of Engineering and Technology
Dhaka, Bangladesh

May 24, 2015

Let's see a problem

Let's see a problem

8 students get following marks in an exam.

Let's see a problem

8 students get following marks in an exam.

29	25	3	49	9	37	21	43

Let's see a problem

8 students get following marks in an exam.

29	25	3	49	9	37	21	43

A teacher needs to sort the marks.

Possible Solution

Possible Solution

We can use some commonly used algorithms here. Like:

Possible Solution

We can use some commonly used algorithms here. Like:

- Bubble Sort

Possible Solution

We can use some commonly used algorithms here. Like:

- Bubble Sort
- Merge Sort

Possible Solution

We can use some commonly used algorithms here. Like:

- Bubble Sort
- Merge Sort
- Quick Sort

Possible Solution

We can use some commonly used algorithms here. Like:

- Bubble Sort
- Merge Sort
- Quick Sort
- etc.

But wait!!!

But wait!!!

Is there a faster way to sort?

But wait!!!

Is there a faster way to sort?

Yes, there is.

BUCKET SORT!!!

Bucket sorting is a linear time algoritm of sorting

BUCKET SORT!!!

Bucket sorting is a linear time algoritm of sorting

Assumptions:

Assumptions:

- inputs are distributed uniformly over a range.

Back to our problem.

Index	0	1	2	3	4	5	6	7
Value	29	25	3	49	9	37	21	43

Back to our problem.

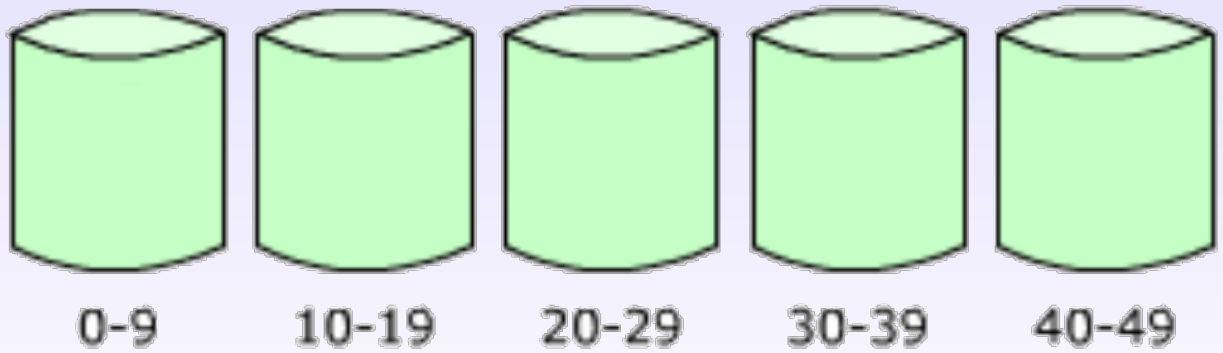
Index	0	1	2	3	4	5	6	7
Value	29	25	3	49	9	37	21	43

Here, we assume

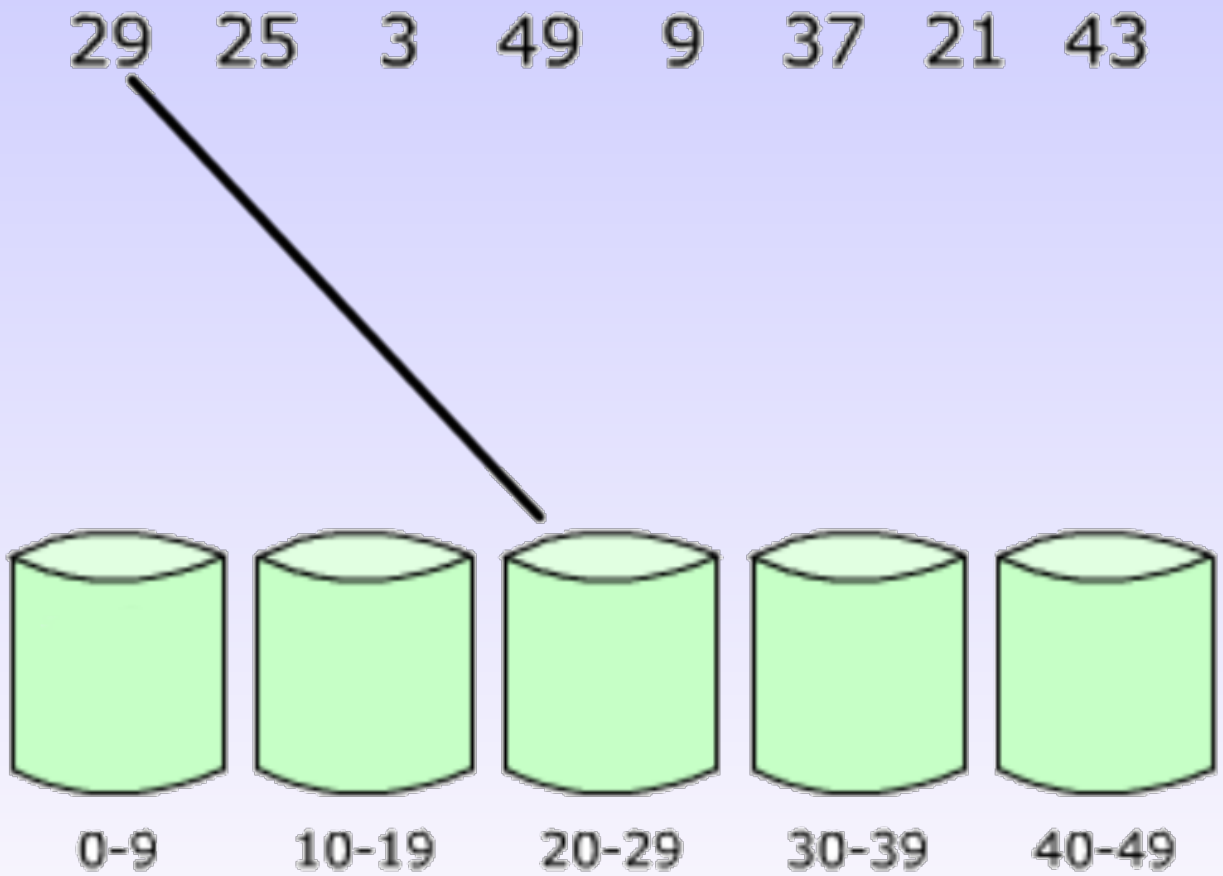
- Array size: 8
- Input range: 0-49
- Let's insert these numbers into 5 buckets.

Sorting...

29 25 3 49 9 37 21 43

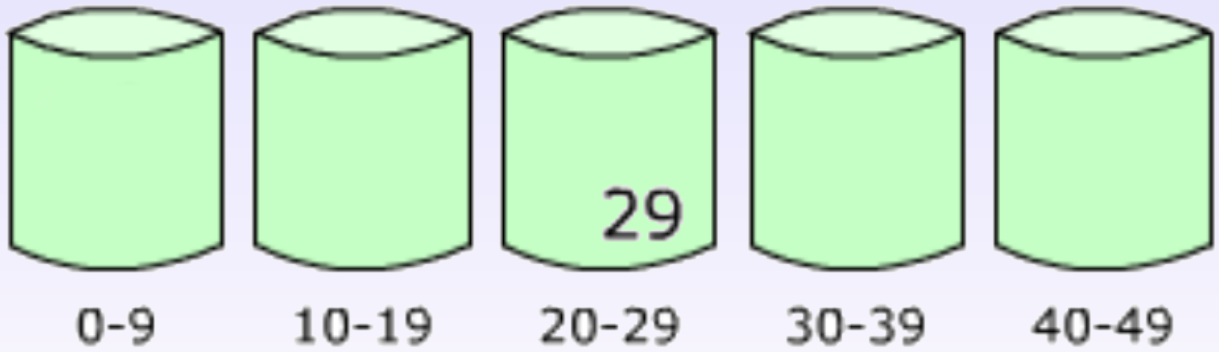


Sorting...

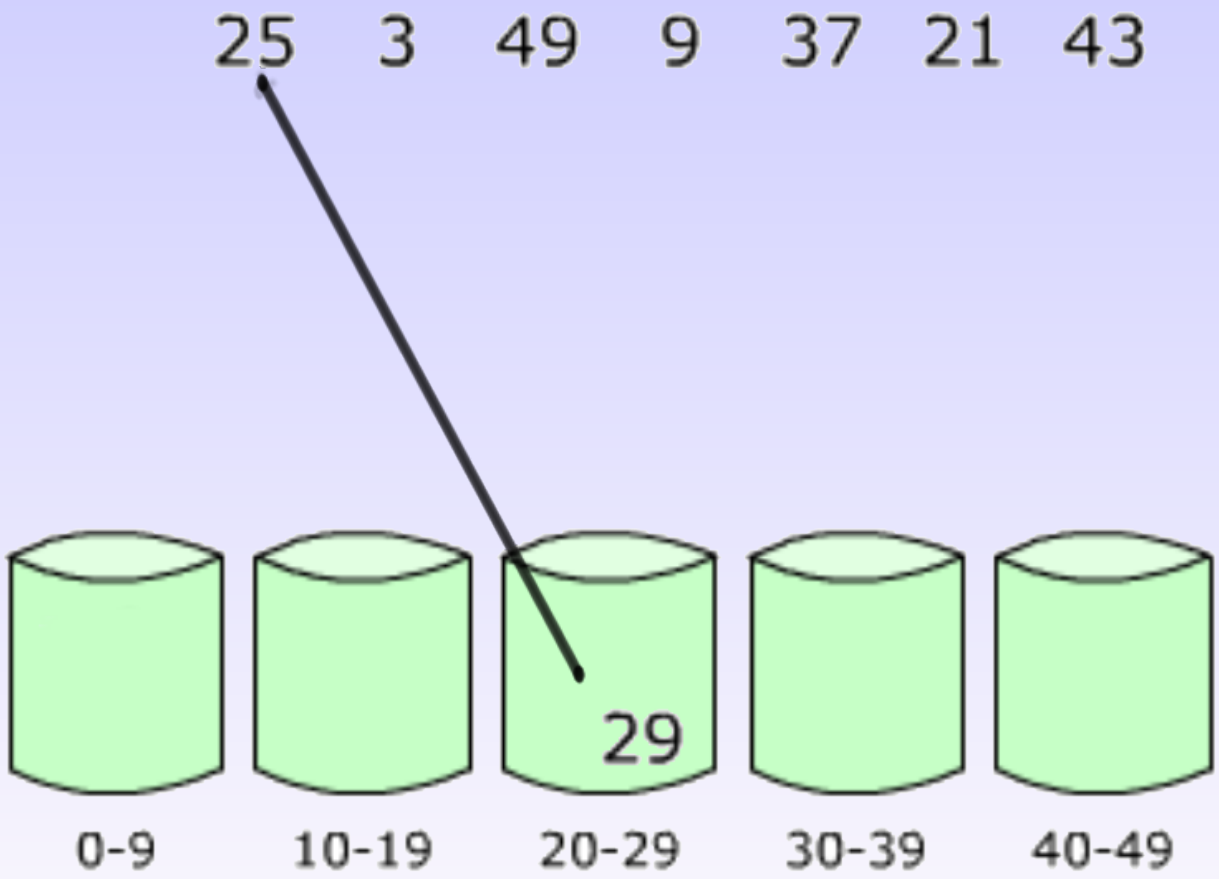


Sorting...

25 3 49 9 37 21 43

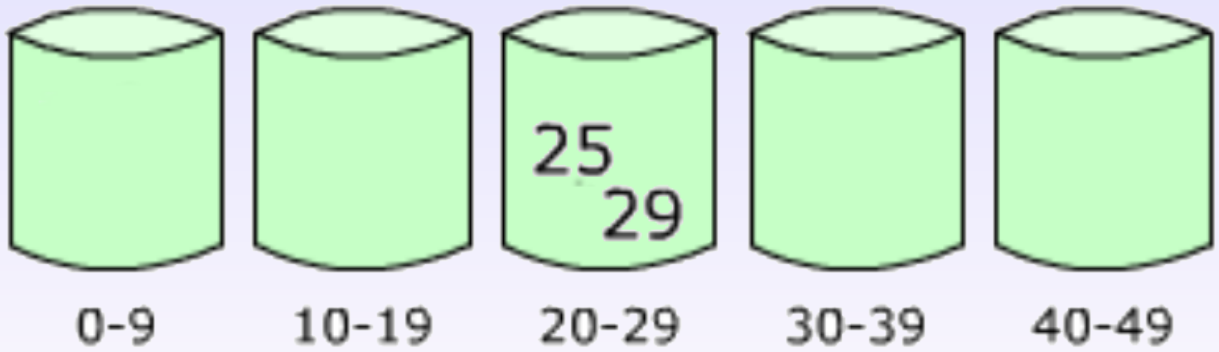


Sorting...

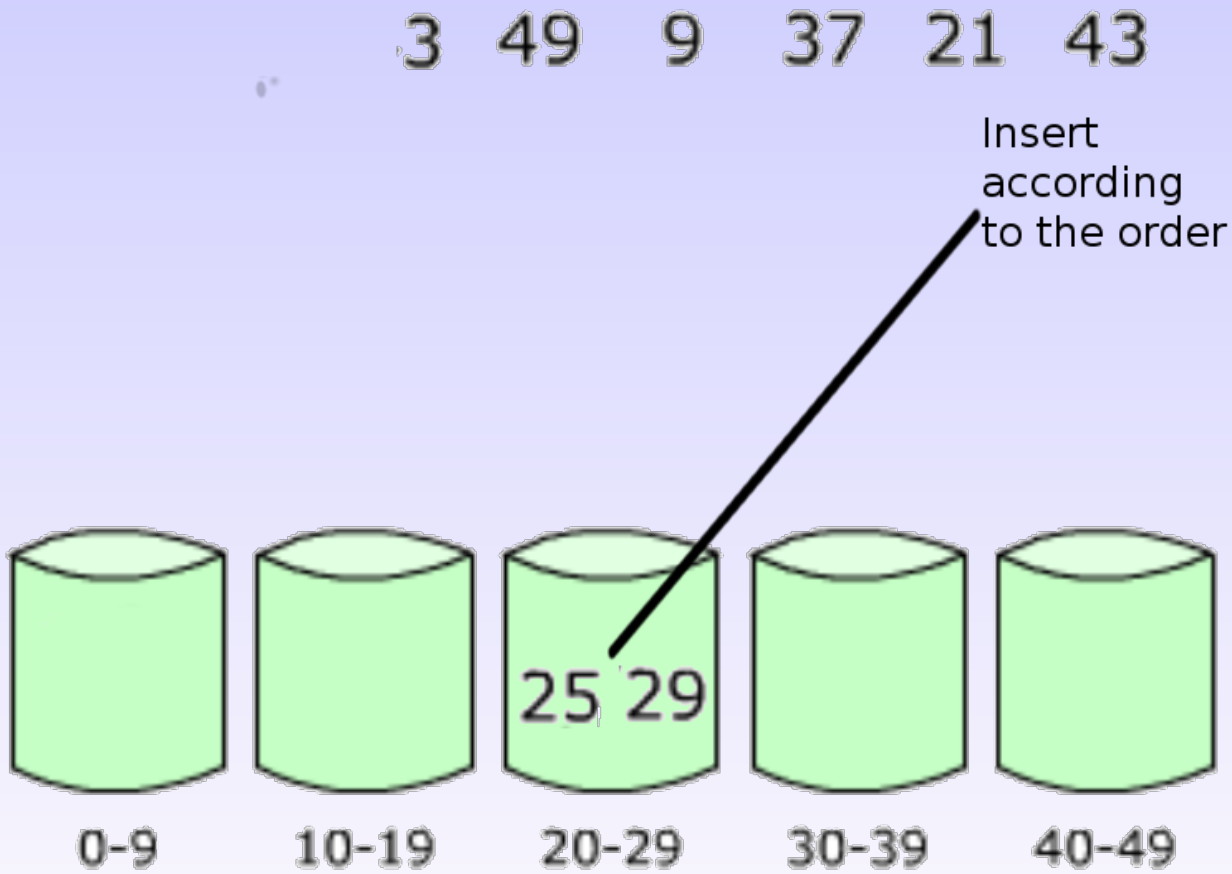


Sorting...

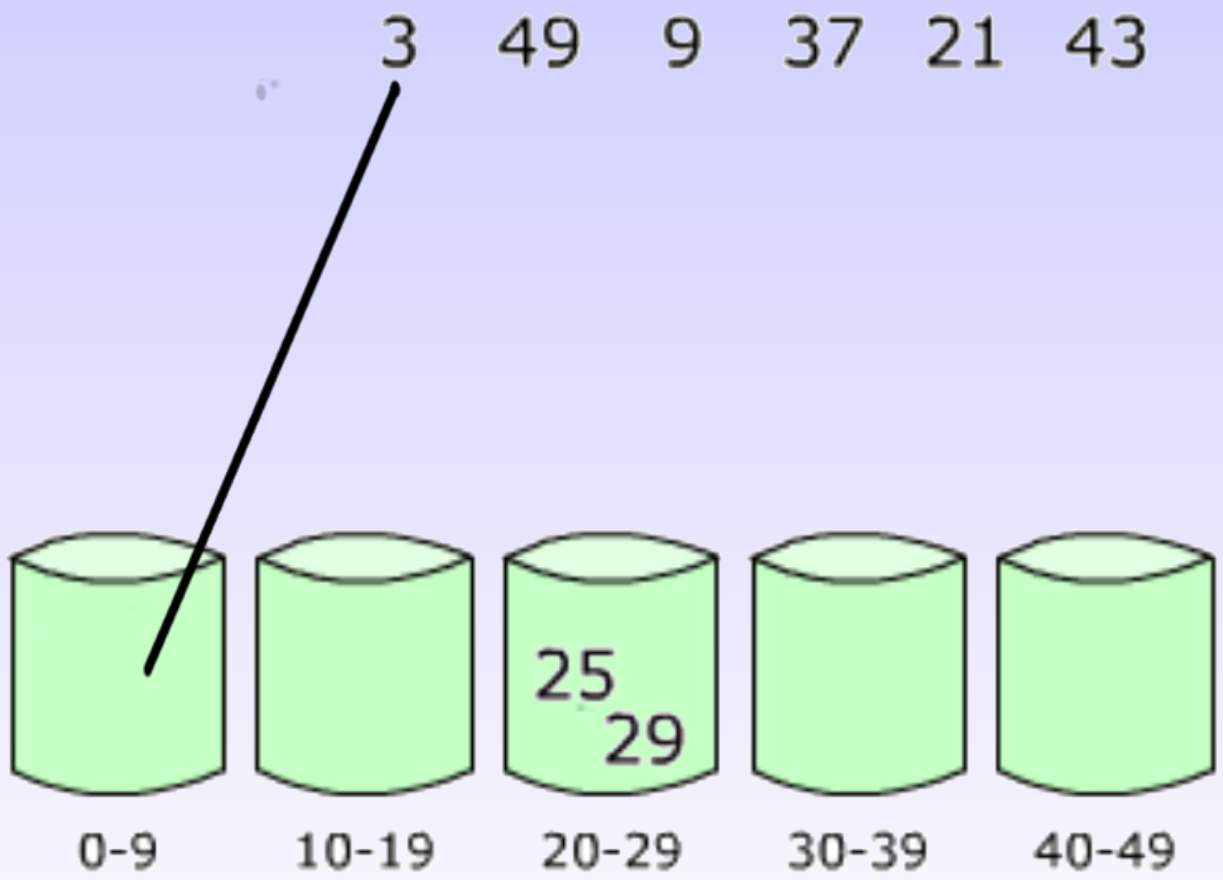
3 49 9 37 21 43



Sorting...

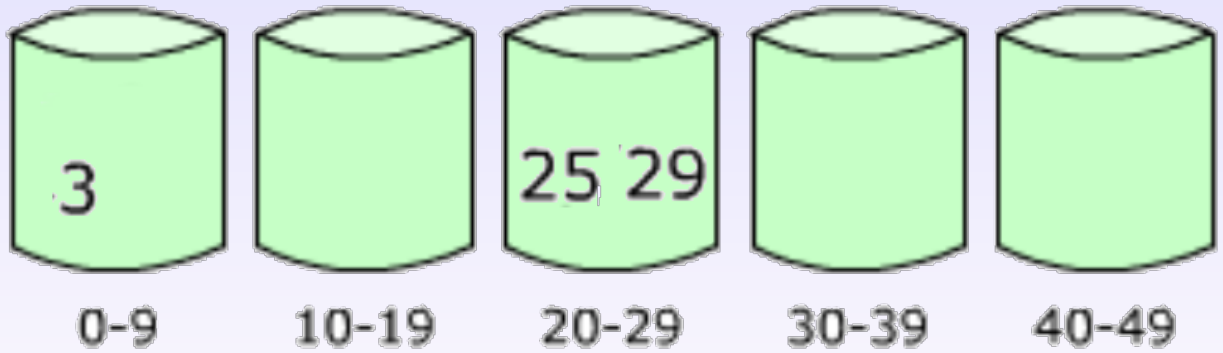

3 49 9 37 21 43
Insert
according
to the order
25 29
0-9
10-19
20-29
30-39
40-49

Sorting...

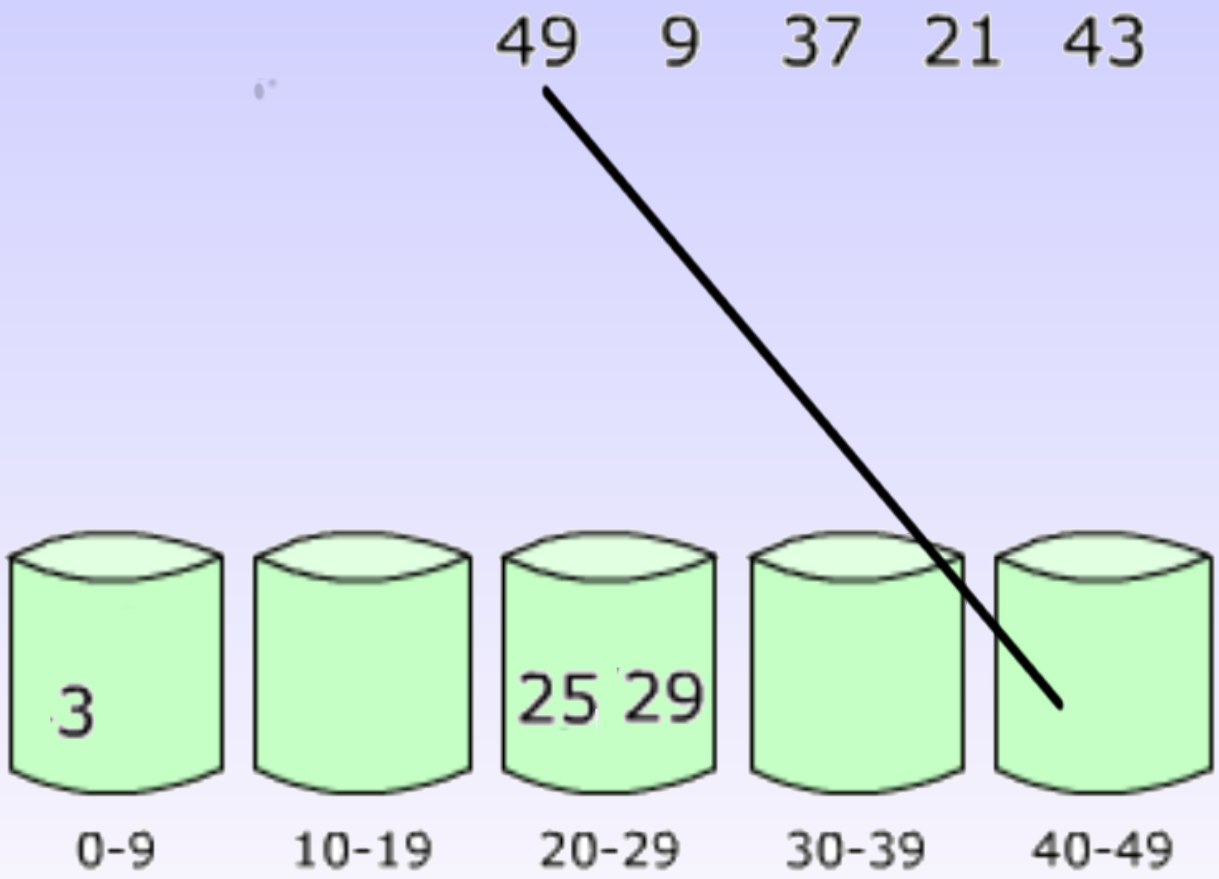


Sorting...

49 9 37 21 43

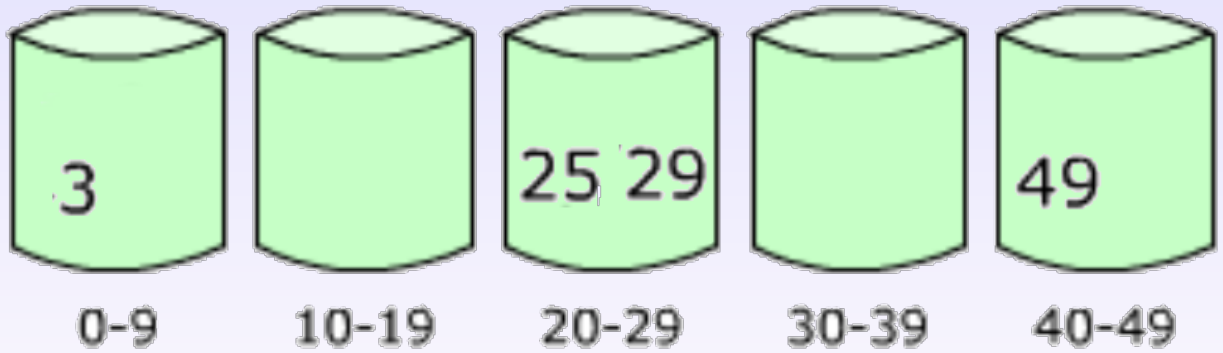


Sorting...

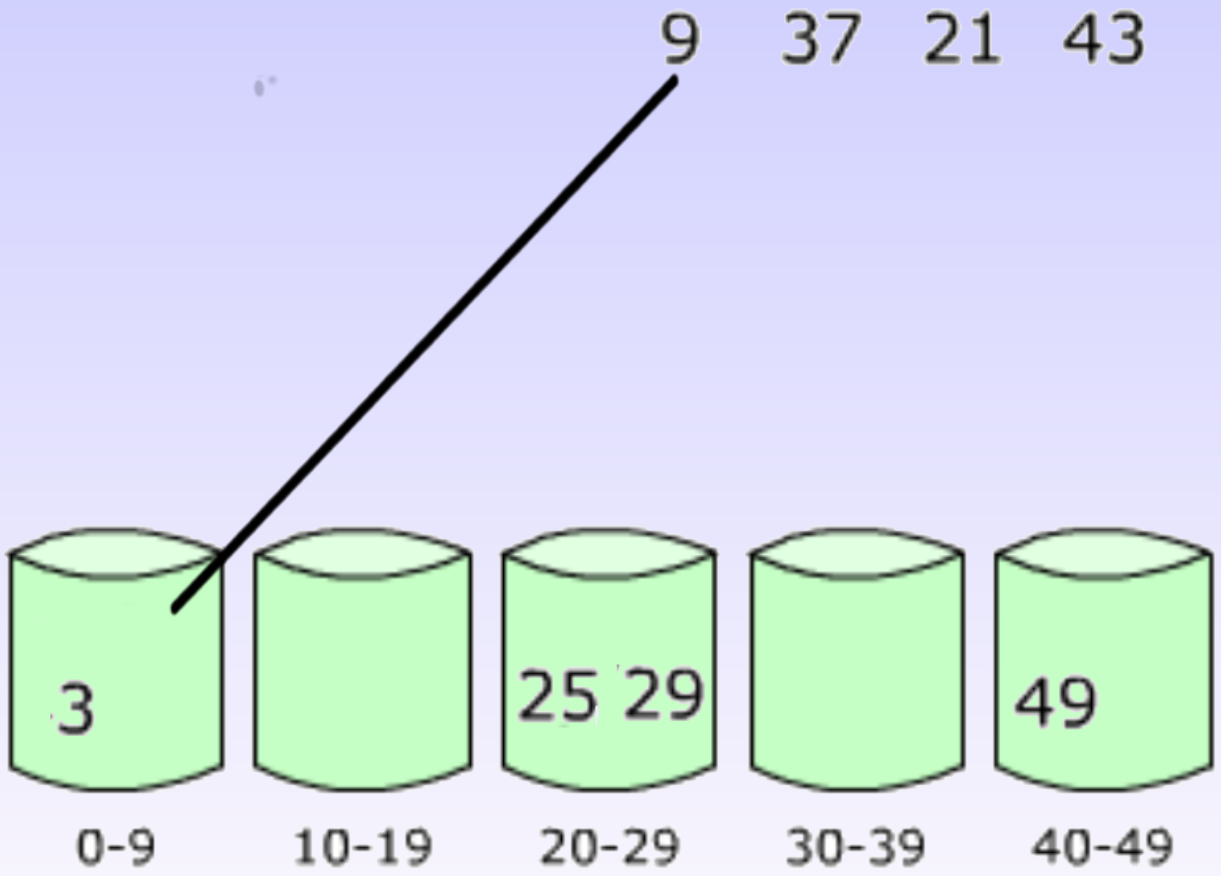


Sorting...

9 37 21 43

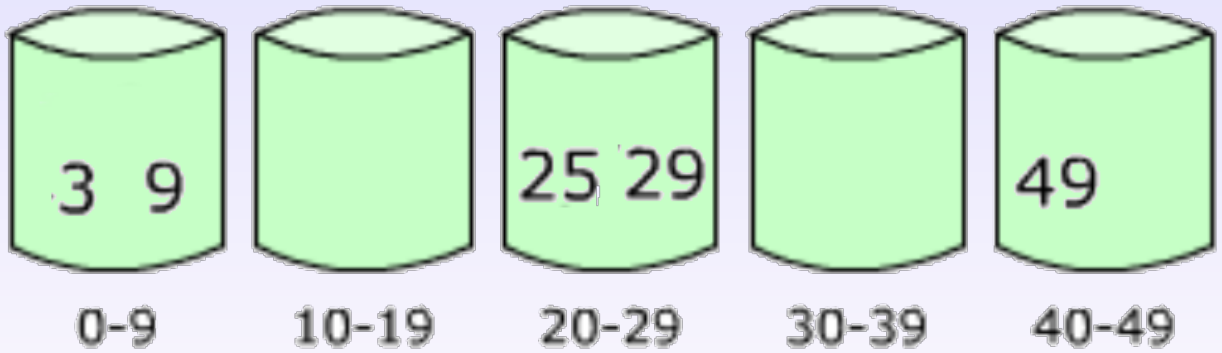


Sorting...

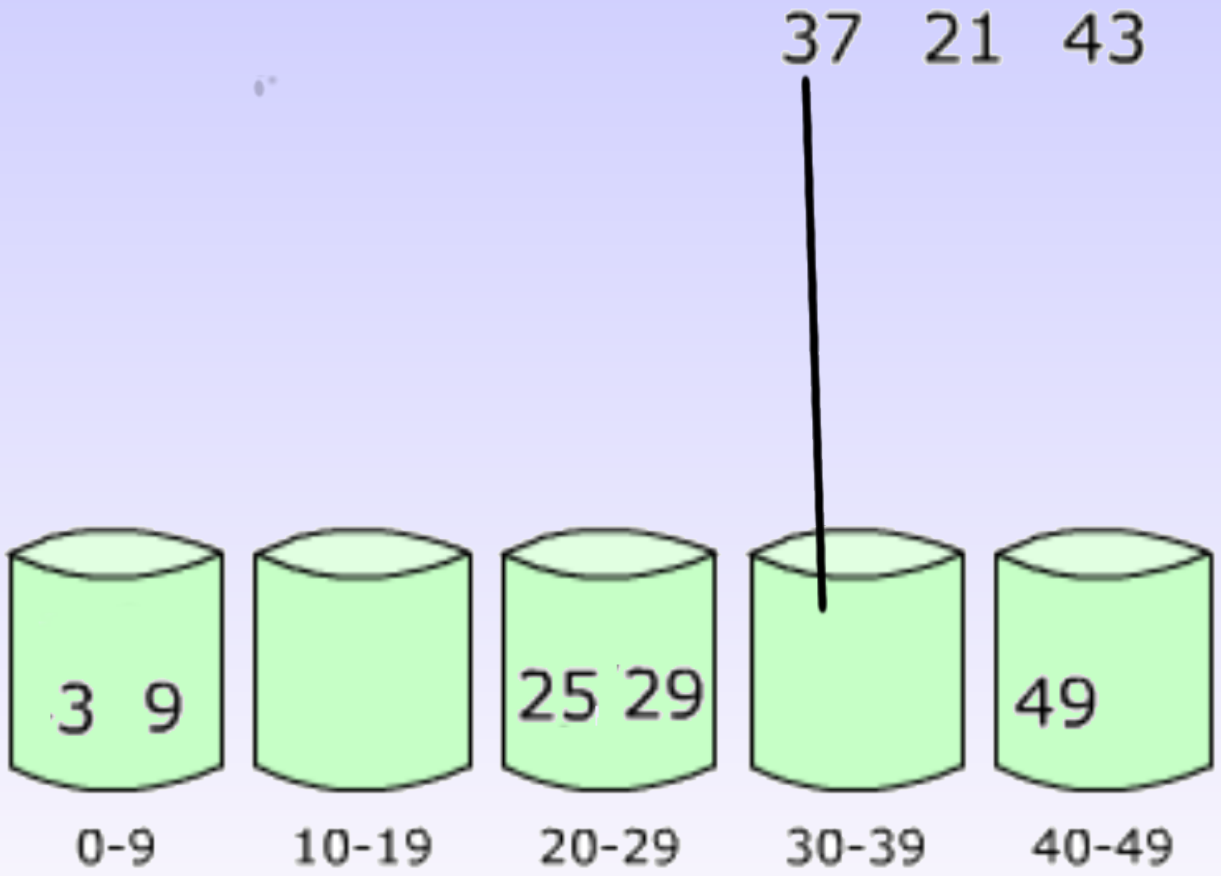


Sorting...

37 21 43

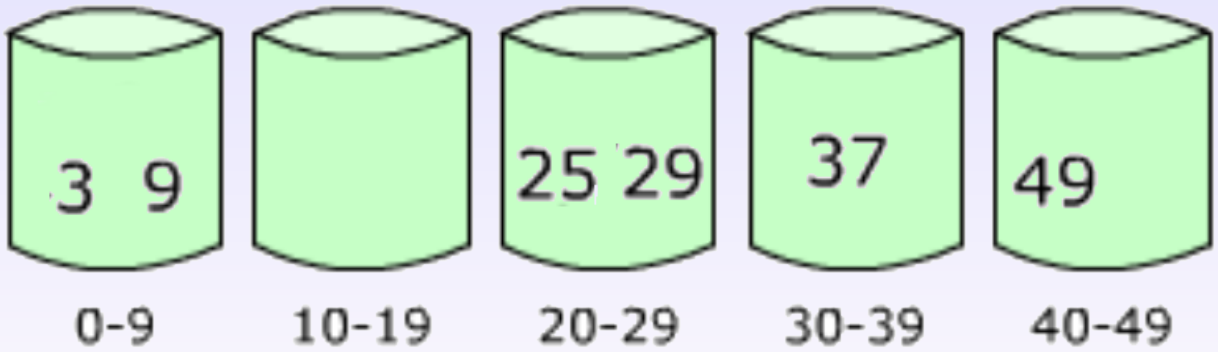


Sorting...

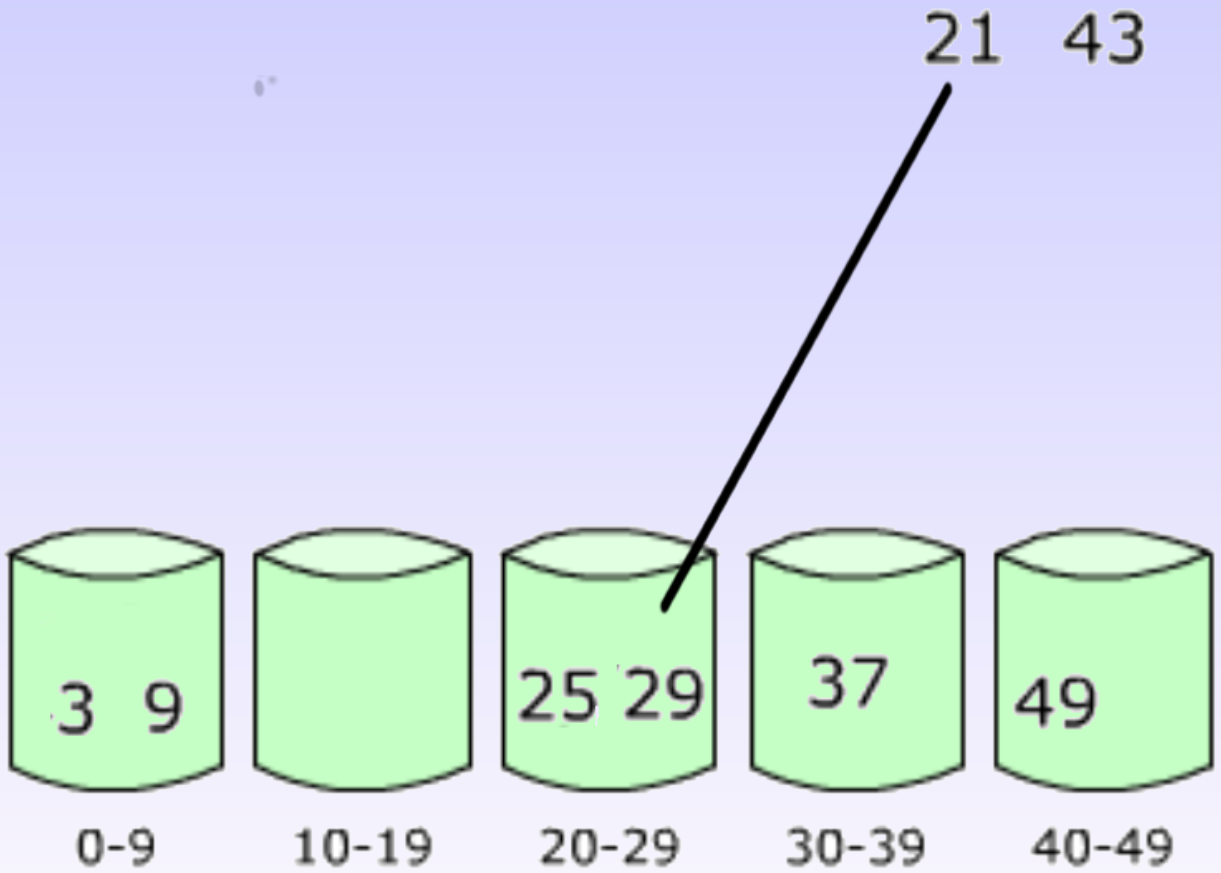


Sorting...

21 43

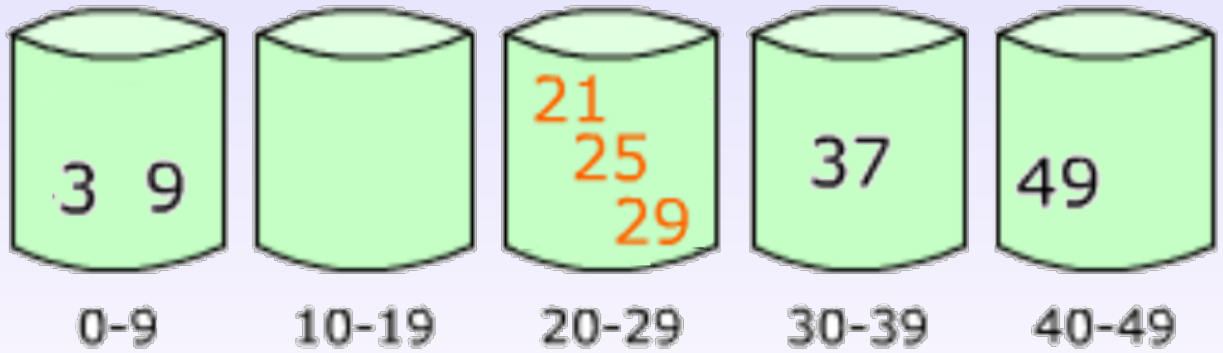


Sorting...

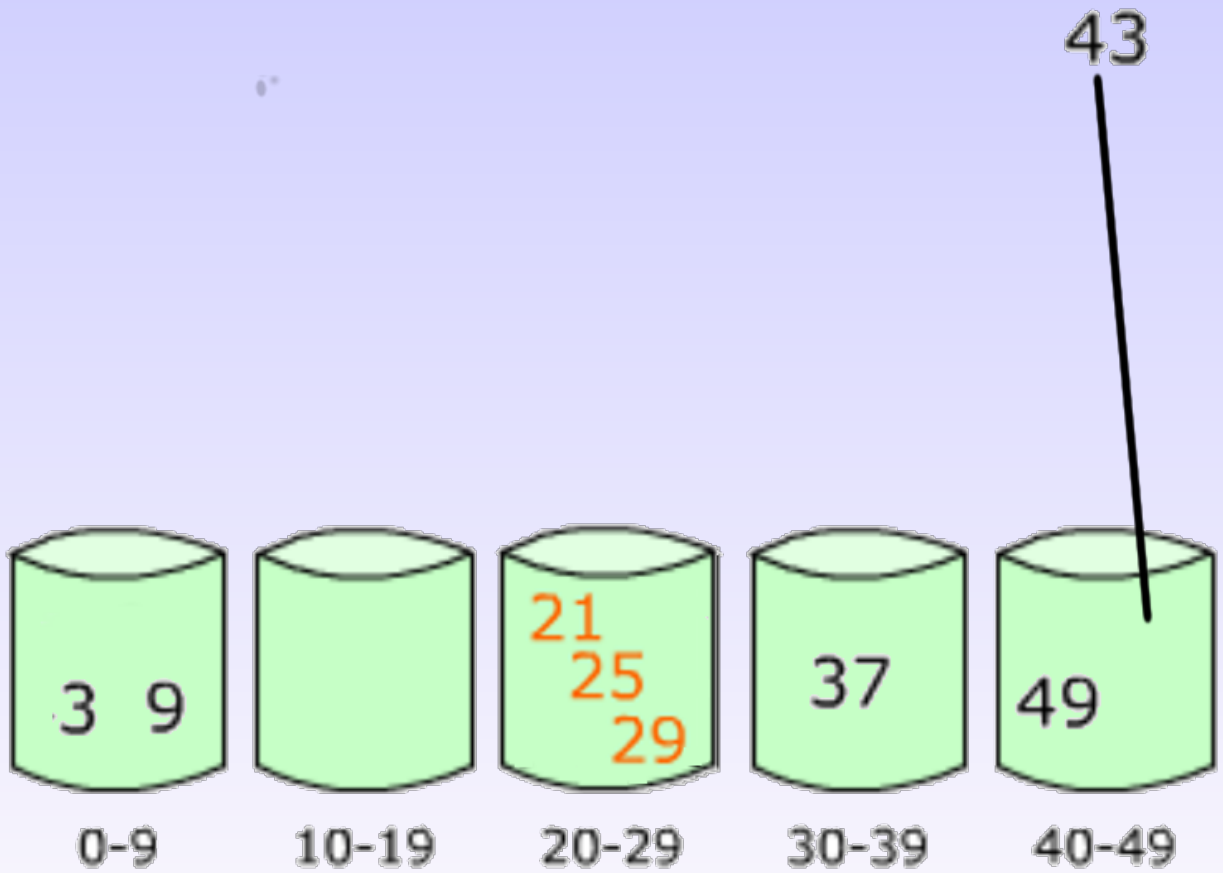


Sorting...

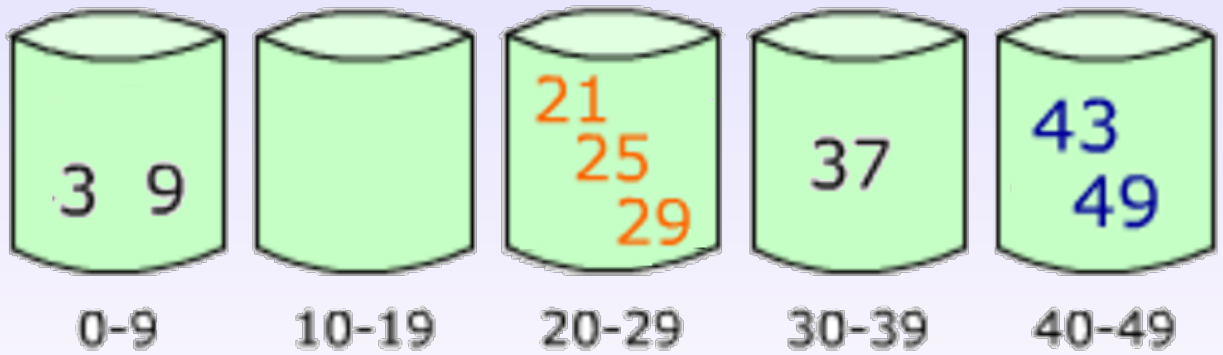
43



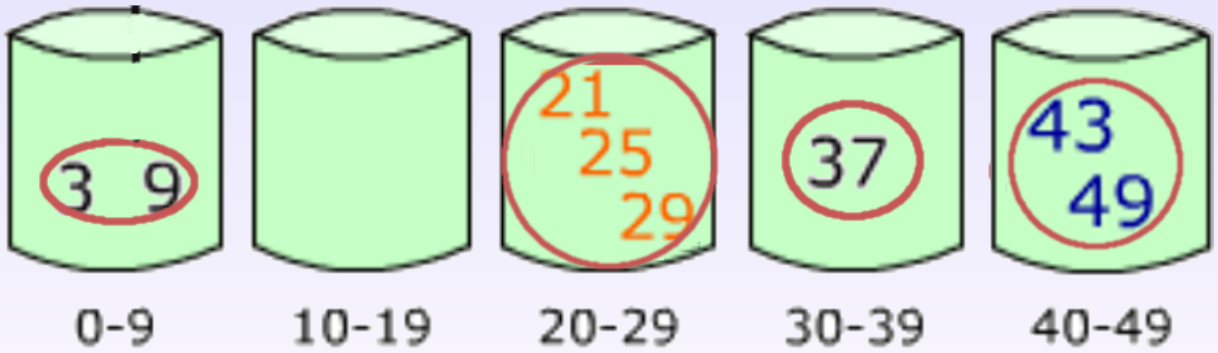
Sorting...



Sorting...

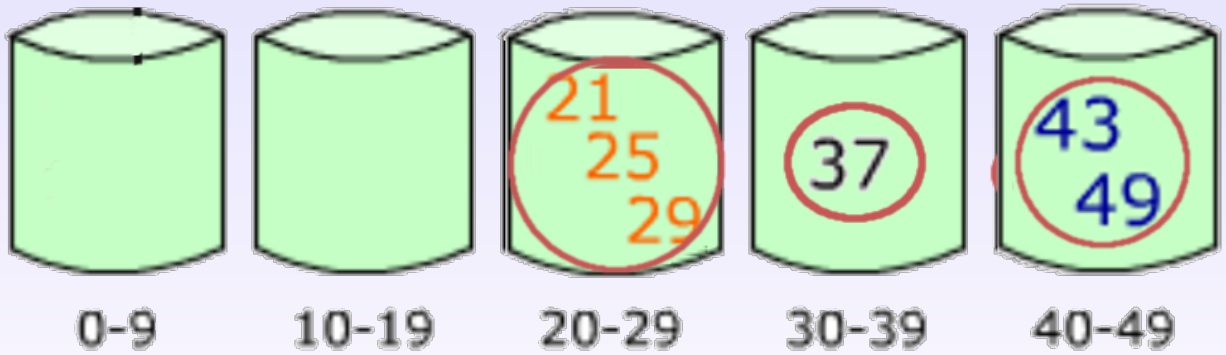


Sorting...



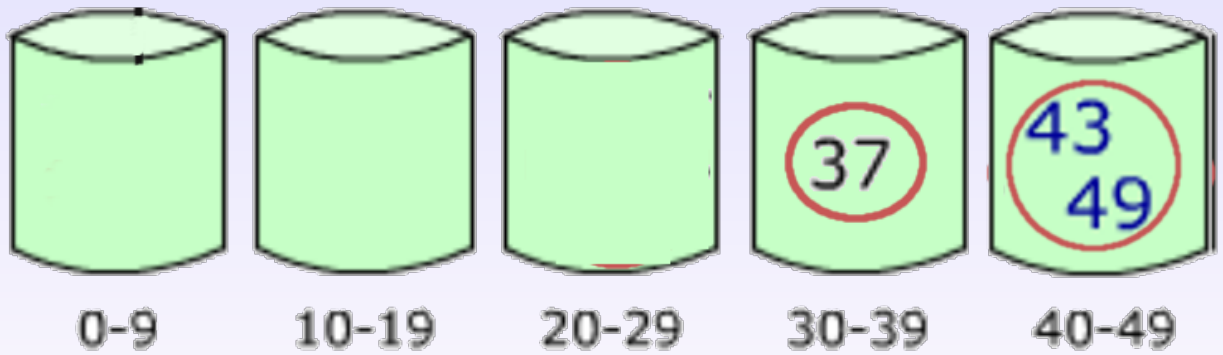
Sorting...

3 9



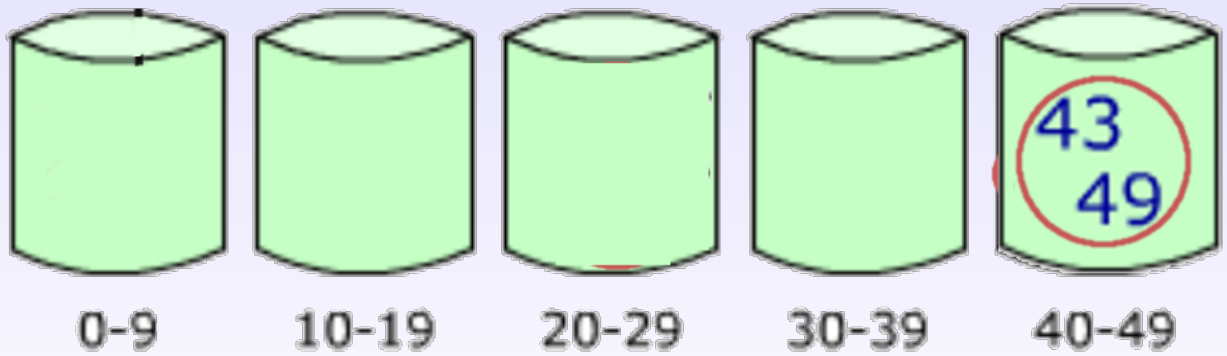
Sorting...

3 9 21 25 29



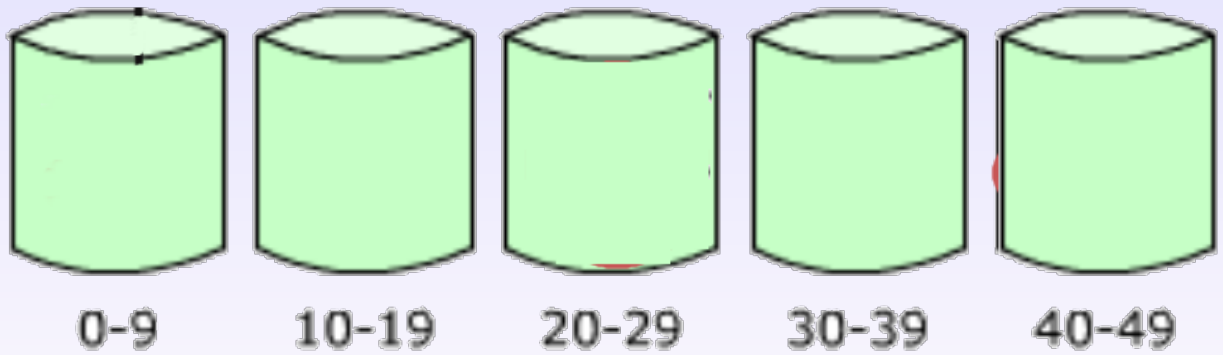
Sorting...

3 9 21 25 29 37



Sorting...

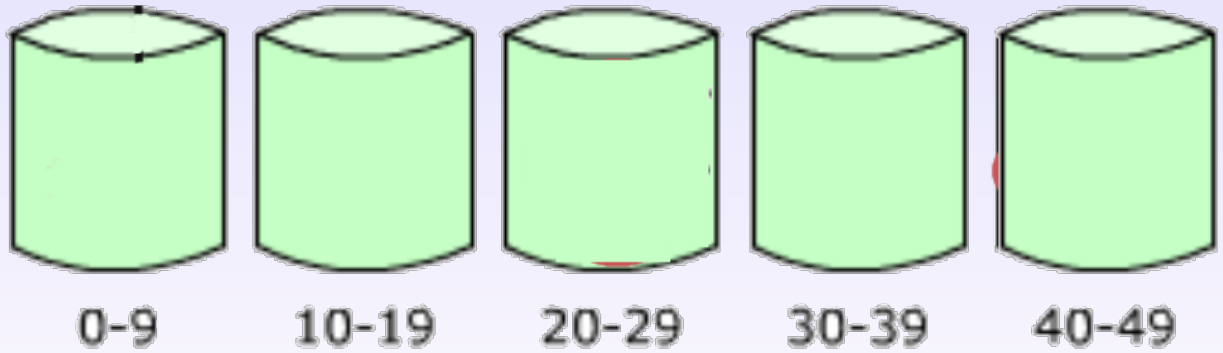
3 9 21 25 29 37 43 49



Sorting...

Sorted Array:

3 9 21 25 29 37 43 49



Brief Overview of Implementation.

Brief Overview of Implementation.

- Take an array of linked list.

Brief Overview of Implementation.

- Take an array of linked list.
- Each element of the array will work as a bucket.

Brief Overview of Implementation.

- Take an array of linked list.
- Each element of the array will work as a bucket.
- Put each number into the appropriate bucket.

Brief Overview of Implementation.

- Take an array of linked list.
- Each element of the array will work as a bucket.
- Put each number into the appropriate bucket.
- Insert each number according to its order.

Brief Overview of Implementation.

- Take an array of linked list.
- Each element of the array will work as a bucket.
- Put each number into the appropriate bucket.
- Insert each number according to its order.
- Merge the buckets.

Issues with bucket sort.

Advantages:

Issues with bucket sort.

Advantages:

- Runtime is proportional to input size.

Issues with bucket sort.

Advantages:

- Runtime is proportional to input size.
- More uniform data takes less time to sort.

Issues with bucket sort.

Advantages:

- Runtime is proportional to input size.
- More uniform data takes less time to sort.
- Technically its expected runtime is $O(n)$.

Issues with bucket sort.

Limitations:

Issues with bucket sort.

Limitations:

- Non uniformly distributed data takes more time.

Issues with bucket sort.

Limitations:

- Non uniformly distributed data takes more time.
- Not suitable for non-versatile data.

References

 *Introduction to Algorithms*, Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein