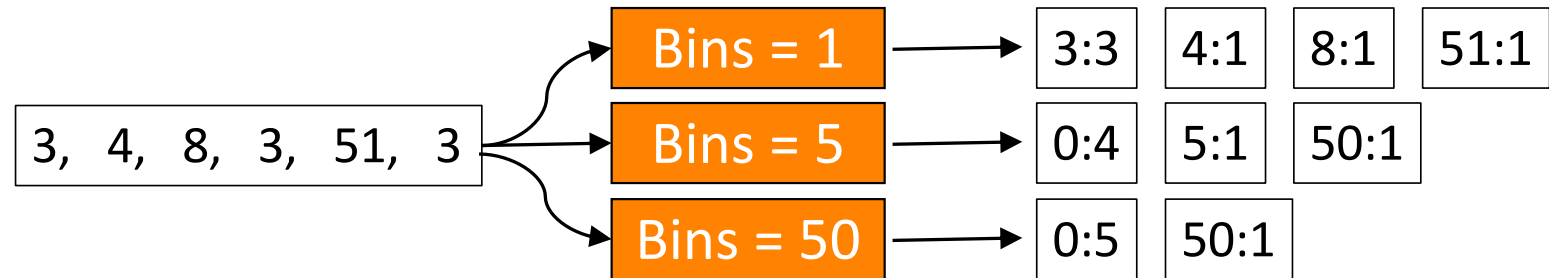


Histogram Functionality

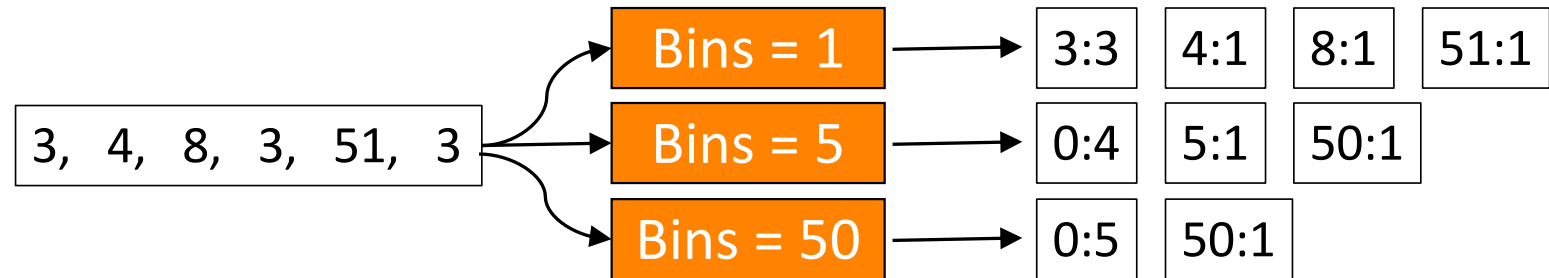


```
bool Set_Bin_Size(double bs);
```

```
bool Add_Value(double d);
```

```
bool Get_Data(vector<int> &bin_ids,  
              vector<int> &num_elts) const;
```

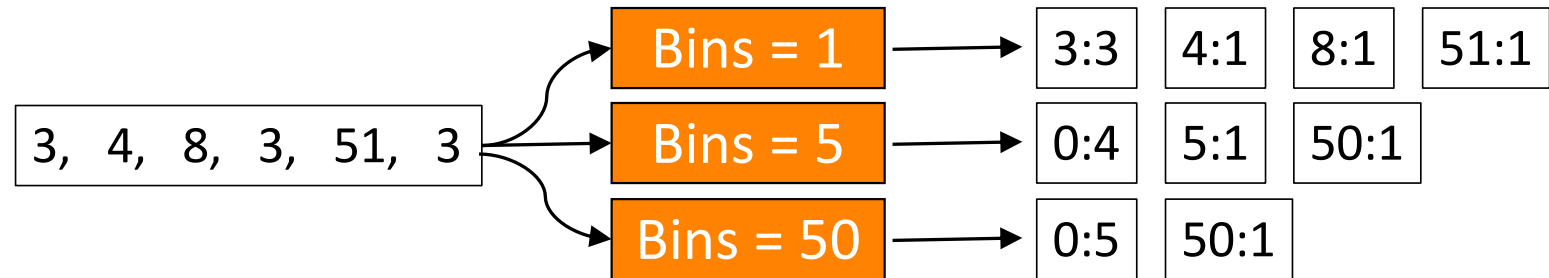
Histogram Functionality



Vector Implementation

XX

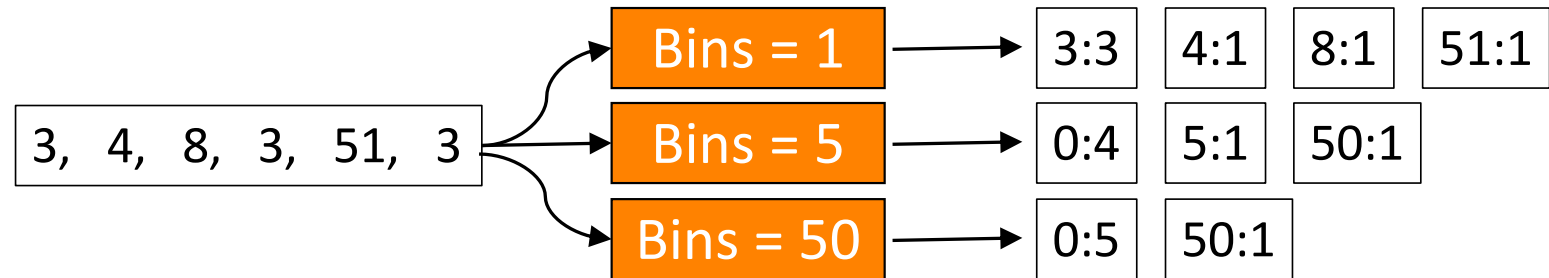
Histogram Functionality



Map Implementation

XX

Histogram Functionality

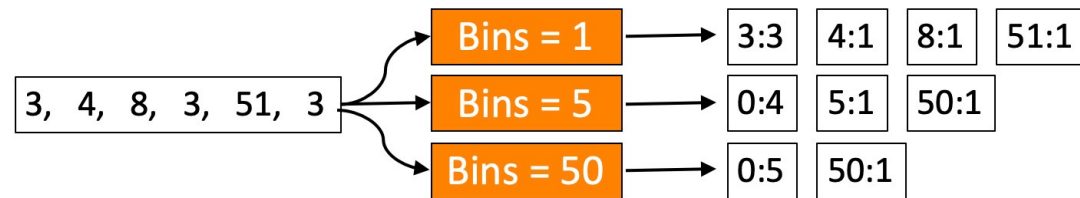


Hash Table Implementation

XX

Metrics:

n: # of elements
bins: # of distinct elements
min: Min bin number
max: Max bin number
ts: Hash table size



Vector Implementation

Vector of ints, for bins 0 through max.

Create: X

Get_Data: X

Space: X

Metrics:

n: # of elements
bins: # of distinct elements
min: Min bin number
max: Max bin number
ts: Hash table size



Vector Implementation

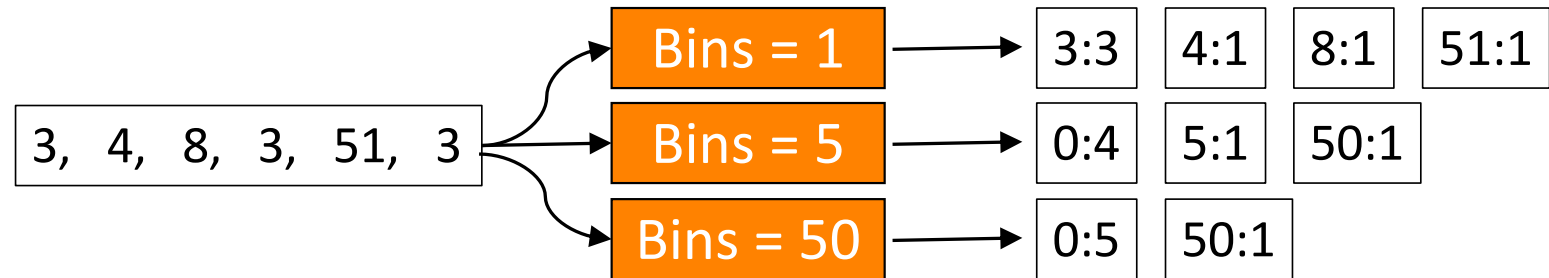
Vector of ints, for bins 0 through max.

Create: $O(n + \max)$

Get_Data: $O(\max)$

Space: $O(\max)$

Histogram Functionality



Map Implementation

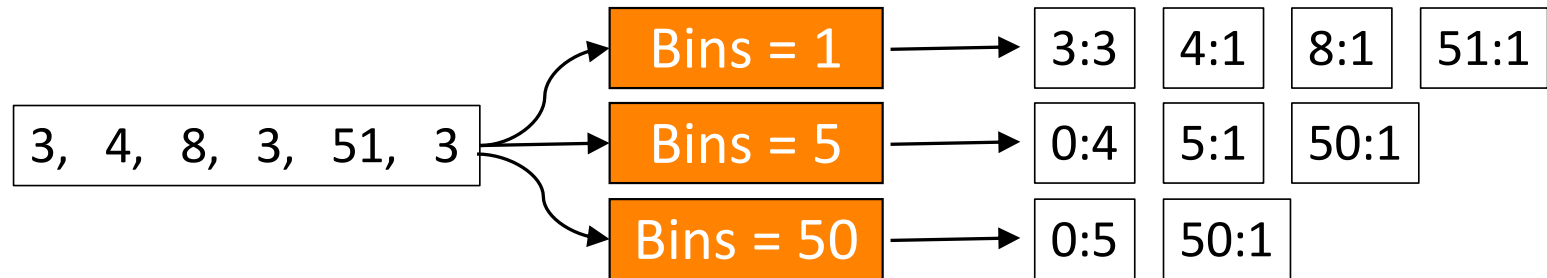
Key = bin, Val = number

Create: X

Get_Data: X

Space: X

Histogram Functionality



Map Implementation

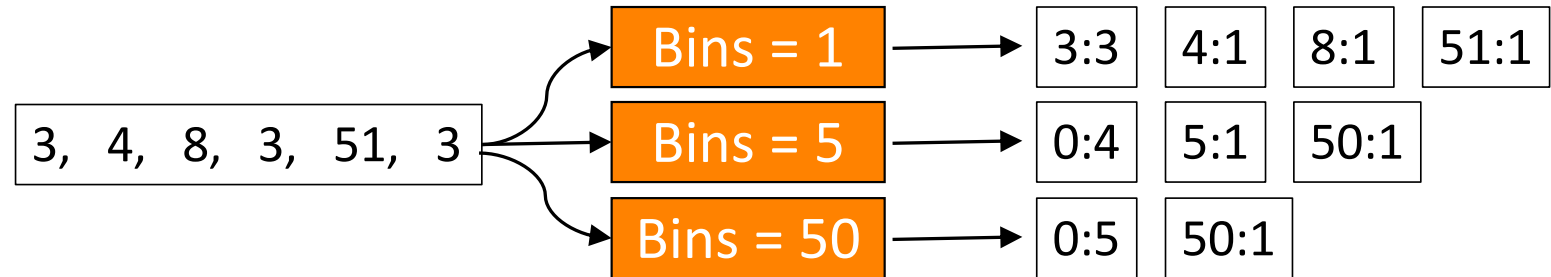
Key = bin, Val = number

Create: $O(n \log bins)$

Get_Data: $O(bins)$

Space: $O(bins)$

Histogram Functionality



Hash Table Implementation

Store keys/vals in a hash table.

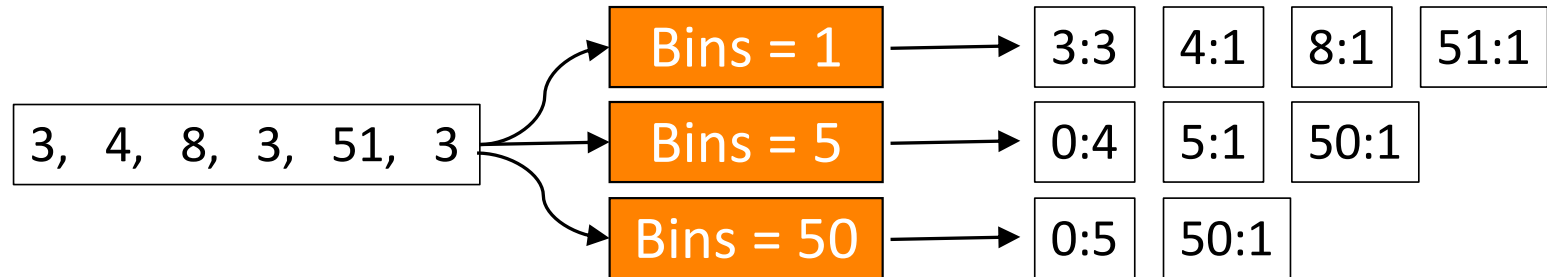
For get_data(), traverse table, put keys into a vector and sort.

Create: X

Get_Data: X

Space: X

Histogram Functionality



Hash Table Implementation

Store keys/vals in a hash table.

For `get_data()`, traverse table, put keys into a vector and sort.

Create: $O(ts + n)$

Get_Data: $O(ts + bins \log bins)$

Space: $O(ts)$