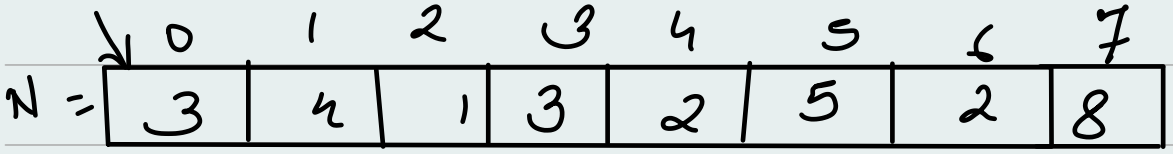
Count Sort Algorithm :-

Count Sort :

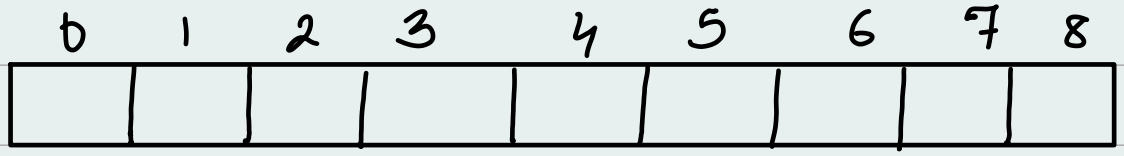
* Non comparison sorting algorithm.
* good for smaller numbers.

O(N+M)

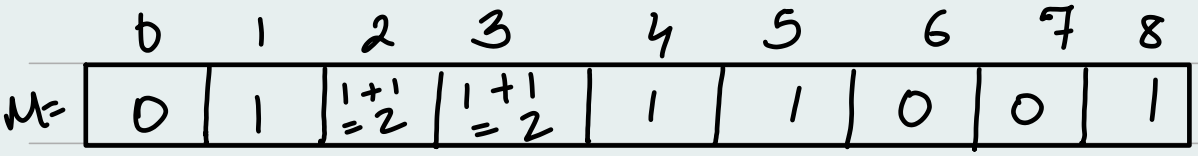


① Select the largest number = 8

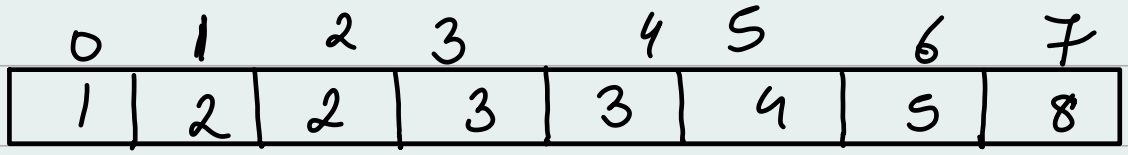
② create an array of largest + 1.
so size 9



③ Now create a frequency array where index no is equal to element number



Replace the original array.



Array Sorted

Time Complexity : O(N+M)
Space complexity : O(N+M)

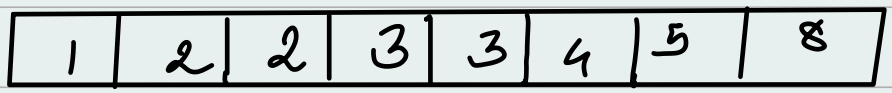
We can also do this by HashMap :

3	2
4	1
1	1
2	2
5	1
8	1

max = 8

i = 0 till 8

map.get(i)



Time complexity : O(N+M)
Space complexity : O(M)

Code :-

```
public class countSort{ //this is a stable sort
                                Algorithm
    public static void countSort(int[] arr){
        if(array == null || array length <= 1){
            return;
        }
        int max = array[0]; // find the max
                              element in the array.
        for(int nums: array){
            if(nums > max){ // if the current
                max = nums; element is greater
            }                 than max
        }

        int[] countArray = new int
                              [max + 1];
// create a count array of size max+1.

        for(int num: array){
            countArray[num]++;
// count the frequency of each element
   in the array
        }
```

```
int index = 0;
for (int i=0 ; i<=max ; i++){
    while (countArray[i] > 0){
        array[index] = i; //
Replace the element in the array with
the sorted element
        countArray[i]--;
    }
}
}
// we can also create countSort using
HashMap to store the frequency of
each element in the array

public static void countSortHashMap(
                    int[] array){

    if (array == null || array.length<=1){
        return;
}
    int max = Arrays.stream(arr).
            max().getAsInt();

    int min = Arrays.stream(arr).
            min().getAsInt();
```

// now MapOut the frequency of each element in the array

```
Map<Integer, Integer> countMap = new
                                HashMap();

    for(int num : arr){
        countMap.put(num, countMap.
            getOrDefault(num, 0)+1);//
If the number is not present, then add
default 0 or else increments the value
by 1
}

    int index =0;

    for(int i = min; i<= max ; i++){
            int count = countMap.getOrDefault
                                (i ,0); //get the
    frequency of the element.

            for(int j =0; j <count; j++){//
loop will run that many times the number appeared
in the array if 2 appeared 2 times it will
run the loop for 2, two times
```

```
            arr[index] = i; //Replace
            index++;
        }
    }
}

public static void main(String[] args){
    int[] arr = {4,2,2,8,3,3,1};

    //countSort(array);
    countSortHashMap(array);

    System.out.println(Arrays.toString(
                                array));
    }
}
```