

Analysis of Run-time of algorithm for problem 1  
is  $O(\log n)$  → which terms of operation is excellent

radix sort → stable

remains stable ...

```
for (int j=0; j<arraySize ; j++) {  
    arr[(a[j]/i)%10]++;  
  
    for(int l=arraySize-1; l>= 0; l--) {  
        out[arr[(a[l]/i)%10]-1]=a[l];  
        arr[(a[l]/i)%10]--;  
  
        for(int i=1; max/i>0; i *= 10) {  
            int out[] = new int[arraySize];  
            int arr[] = new int[10];  
            for(int x=0; x<arr.length;x++) {  
                arr[x]=0;  
            }  
        }  
    }  
}
```

// bottleneck is where int [out] and size of array is initialize and created

major change of using  $O(n)$  extra space would have a little worse time complexity then the radix sort, but if not using constant extra space and using extra space provide more time complexity and runtime analysis but this time it specific to value of 1