

1.

The runtime for this problem is $O(n \log_{10} n) = O(n \log n)$

The **bottleneck** that we can identify in the code is

```
for (int x = n - 1; x >= 0; x--) {  
    out[count[ (array[x]/e)%10 ] - 1] = array[x];  
    count[(array[x]/e)%10]--;  
}
```

This part of the code puts the array to be compared for next step in the sorting algorithm.

The sequence is being sorted on the $(x-1)$ th loop and the lower $(x-1)$ th digit for the interval 2×10^x .

The sort keeps these numbers in the same order of the original array. The sort is stable because the order is being maintained. The run time is of $O(n \log n)$, so it already has the extra spacing location. Because we are swapping values and sorting them the sequence of numbers are going to be updated, so the input is going to eventually be replaced by a new output. We are not adding any additional data so it makes this method fast.