Jessica Chen

CSCI 303: Algorithms, HW 9

Due: 3:00 pm, Friday, 11/2

1. Problem 1

Digits	42	57	7	40	83	78	86	89	80	91	79	84
0	40	80										
1	91											
2	42											
3	83											
4	84											
5												
6	86											
7	57	07										
8	78											
9												
	40	80	91	42	83	84	86	57	07	78		
0	07											
1												
2												
3												
4	40	42										
5	57											
6												
7	78											
8	80	83	84	86								
9	91											
	07	40	42	57	78	80	83	84	86	91		

There are 12 enqueues, then 12 dequeues. It then repeats again with 12 enqueues, then 12 dequeues to finish the sort.
Total assignments: $12+12+12+12 = 4*12 = 48$
The total number of movements is a k ($k = 4$) times n for the radix sort no matter how array is originally where as in other sorts the time complexity could be better than radix sort's time complexity, or worse based on how the array is originally ordered. Compared to other's we have compared for to with the same input this is much faster. (i.e. quick sort, heap sort) Radix is better because it needs less assignments than a swap, which requires at least 3 assignments to do a swap, and it does not need to shift anything, which requires as many assignments as items that need to be shifted.

2. Problem 2
Counting sort is not as efficient space wise because the sequence of integers is over a large range so the count array it is put into is large, but it is efficient in respect to time because it will only need to iterate through the array at most k times, where k is the maximum value of the key.