Computer Science II	Lab 19D – RADIX SORT	Lab Value - 100

Lab Goal : This lab was designed to teach you more about sorting and searching and specifically about the radix sort.

How does a Radix Sort work?

The first step in writing a radix sort is to find the number of digits in the biggest number. You need this information because a radix sort will make passes through a list of numbers starting with the least significant digit or 1s place and then working up to the most significant digit. If the biggest number in the list was 15, the radix sort would have to make two passes to sort the numbers as it would have to sort by the 1s digit and then by the 10s digit. If the biggest number in the list was 1234, the radix sort would have to make 4 passes to sort all of the numbers. Collections.max() might prove useful for this lab.

For this lab, our number system or base will be 10. Radix sort will start the first pass by sorting the numbers based on the 1s position or least significant digit. During this first pass, radix sort will examine the digit of each number in the 1s position and based on the 1s position, put each number in the queue/list that it matches. If the first number was 56, radix sort would put 56 in the queue 6. If the second number was 48, radix sort would put 48 in queue 8. After putting all numbers in the list in the appropriate queue, radix sort will start at queue 0 and dump the numbers back into the list. Radix sort will move from queue 0 to queue 1 and continue this process through all 10 queues. After dumping all queues, radix sort will move to the 10s digit and start the process all over again. This will continue for the number of digits in the largest number.

Radix sort is most commonly implemented with an array of queues, but you could use other structures.

What does radix mean?

A radix refers to the number system/number base in which you are working. In this program we are working with base 10. If were working in base 16, we would have to have an intermediate array of size 16 to handle all of the digit possibilities.

Radix Sort Examples

Original List { 103 , 17 , 38 , 20 }

Pass 1 - Orders #s based on 1s digit

Put #s in array of queues
0 - 20
1 -
2 -
3 - 103
4 -
5 -
6 -
7 - 17
8 - 38
9 –

Copy back to List
{ 20 , 103, 17, 38 }

Original List { 20 , 103, 17, 38 }

Pass 2 - Orders #s based on 10s digit

Put #s in array of queues
0 - 103
1 - 17
2 - 20
3 - 38
4 -
5 -
6 -
7 -
8 -
9 –

Copy back to List
{ 103 , 17, 20, 38 }

Original List { 103 , 17, 20, 38 }

Pass 3 - Orders #s based on 100s digit

Put #s in array of queues
0 – 17 20 38
1 - 103
2 -
3 -
4 -
5 -
6 -
7 -
8 -
9 –

Copy back to List
{ 17 , 20, 38, 103 }

Sample Output :

```
Original List -  [103, 17, 38, 20]

Sorted List -  [17, 20, 38, 103]


Original List -  [655, 21, 22, 48, 262, 6412, 11, 33, 10]

Sorted List -  [10, 11, 21, 22, 33, 48, 262, 655, 6412]
```

Why can we use an array of size 10?

Why do you not need an array of bigger size?