CSCA48 Exercise 9

Due: March 24, 2017. 5:00pm

This week, we'll been discussing sorting algorithms in lecture. Let's try an interesting algorithm that we WON'T be discussing.

Radix Sort

Radix sort is a special sort that works on non-negative base 10 integers. Radix sort is implemented using a series of containers or bins: one main bin and 10 digit bins.

Each bin is implemented as a First in First out (FIFO) collection. The algorithm follows these steps:

1. Start by placing all of the numbers in the main bin.
2. Remove each value in the main bin and place it in a digit bin corresponding to the digit being considered, starting with the least significant digit.
3. Once all the values are placed in the appropriate digit bin, collect the values from bin 0 to bin 9, in ascending order, and place them back in the main bin.
4. Repeat steps 2 and 3 with the tens digit, the hundreds digit, and so on. After the last digit is processed, the main bin will contain the values in ascending order.

Example

Let's say we have a list that contains the following integers: 240, 28, 5, 18, 140, 2

Your radix sort will place all six numbers in the main bin.

240	28	5	18	140	2

Next, consider the 1's (least significant) digit for each number in the main bin and place it in its corresponding digit bin. After the first pass, the digit bins will look like this:

0-bin	1-bin	2-bin	3-bin	4-bin	5-bin	6-bin	7-bin	8-bin	9-bin
240		2			5			28	
140								18	

Now, in ascending order, place these integers back into the main bin. The main bin now looks like this:

240	140	2	5	28	18

Next, consider the 10's digit for each number (in ascending order) and place it in its corresponding digit bin. When we consider the 10's digit for the number ”2”, it may be helpful to think of it as the number ”02”.

In our example example, after this second pass the digit bins will look like this:

0-bin	1-bin	2-bin	3-bin	4-bin	5-bin	6-bin	7-bin	8-bin	9-bin
2	18	28		240					
5				140					

Now, in order, we place these integers back into the main bin. The main bin now looks like this:

2	5	18	28	240	140

Next, we consider the 100's digit for each number (in order) and place it in its corresponding digit bin. For example, after this pass the digit bin will look like this:

0-bin	1-bin	2-bin	3-bin	4-bin	5-bin	6-bin	7-bin	8-bin	9-bin
2	140	240							
5									
18									
28									

Now, in order, we place these integers back into the main bin. The main bin now looks like this:

2	5	18	28	140	240

If we tried the 1000's digit we would find that all the numbers would go into the 0-bin. We are done and the list is sorted.

What to Submit

In a file called `ex9.py` define a function `radix_sort` that takes a list of integers and returns a list with the same integers sorted in non-decreasing order. Your code only needs to work with non-negative integers. You may **not** use any built-in sorting functions (obviously).

Extra

Here's something to think about... what is the complexity of radix sort? You may have heard that sorting is always at leat $O(n * log(n))$, but this seems to be better than that, isn't it? The clearest (correct) original (not researched from the internet) explanation posted on Piazza will get a challenge point.